ANALYSIS OF A VERTEBRATE RETINA MODEL

THESIS

Stephen M. Wiener, BSEE
Captain, USAF

AFIT/GE/ENG/90S-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/90S-01

ANALYSIS OF A VERTEBRATE RETINA

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of ıechnology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Stephen M. Wiener, BSEE
Captain, USAF

August 1990

ACKNOWLEDGEMENTS

I would like to thank my thesis committee, especially my thesis advisor, Dr. Matthew Kabrisky, for the guidance and support I received in completing this project. I also want to thank Mr. Dan Zambon for helping me with the computer system at AFIT. Finally, I want to acknowledge those individuals at Eglin AFB who supported me in this project, especially Maj Paul Sandburg, Capt Peter Mahon, Mr. Gary Johnson and Mrs. Prudence Washington.

TABLE OF CONTENTS

Acknowledgements.................................... ii

List of Figures iv

List of Tables vii

Abstract viii

I. Introduction 1

A. Background 1
B. Problem 1
C. Scope 2
D. Approach 3

II. Retina Functions 4

III. Experimental Design 10

A. Equations for the Output of Retina Cells . 10
B. Input Variables 12
C. Program Implementation 14

IV. Results and Analysis 16

A. Introduction 16
B. Edge Enhancement 16
C. Shape Filtering 42

V. Conclusion 69

VI. Recommendations for Further Research 71

Appendix: Program Listing 73

Bibliography 83

Vita 84

LIST OF FIGURES

Figure 1 Retina Cell Connections 5

Figure 2 Illustration of Lateral Connection Effects ... 8

Figure 3 Plot of Equation 3-2 11

Figure 4 Case 1 Noise-Free Rectangle 20

Figure 5 Case 2 Edge-enhanced Noise-free Rectangle .. 21

Figure 6 Case 3 Edge-enhanced Noise-free Rectangle, Increased Exponent 22

Figure 7 Case 4 Edge-enhanced Noise-free Rectangle, Increased Exponent 23

Figure 8 Case 5 Edge-enhanced Noise-free Rectangle, Larger Inhibition Region 24

Figure 9 Case 6 Edge-enhanced Noise-free Rectangle, Large Excitation and Inhibition Region 26

Figure 10 Case 7 Edge-enhanced Noise-free Rectangle, Smaller Inhibition Region 27

Figure 11 Case 8 Edge-enhanced Rectangle Without Additional Increment 29

Figure 12 Case 9 Vertical Bars 30

Figure 13 Case 10 Edge-enhanced Vertical Bars 31

Figure 14 Case 11 Edge-enhanced Vertical Bars, Increased Exponent 32

Figure 15 Case 12 Edge-enhanced Vertical Bars, Increased Exponent 33

Figure 16 Case 13 Edge-enhanced Vertical Bars, Larger Inhibition Region 35

Figure 17 Case 14 Edge-enhanced Vertical Bars, Smaller Inhibition Region 36

Figure 18 Case 15 Rectangles in Noise 37

Figure 19 Case 16 Processed Rectangles in Noise 38

LIST OF FIGURES (CONTINUED)

Figure 20 Case 17 Processed Rectangles in Noise, Increased Exponent 39

Figure 21 Case 18 Processed Rectangles in Noise, Increased Exponent 40

Figure 22 Case 19 Processed Rectangles in Noise, Large Inhibition Region 41

Figure 23 Case 20 Processed Rectangles in Noise, Larger Inhibition Region 43

Figure 24 Case 21 Processed Rectangles in Noise, Small Excitation and Inhibition Regions ... 44

Figure 25 Case 22 Processed Rectangles in Noise, Large Excitation Region 45

Figure 26 Case 23 Processed Rectangles in Noise, Large Excitation and Inhibition Regions ... 46

Figure 27 Case 24 Processed Rectangles in Noise, Large Inhibition Region, Large Exponent ... 47

Figure 28 Case 25 Processed Rectangles in Noise, Small Excitation and Inhibition Regions, Large Exponent 48

Figure 29 Case 26 Three Squares 50

Figure 30 Case 27 Three Squares, Filtered 51

Figure 31 Case 28 Three Squares, Filtered, Larger Inhibition Region 52

Figure 32 Case 29 Three Squares, Filtered, Small Exponent 53

Figure 33 Case 30 Orthogonal Rectangles 55

Figure 34 Case 31 Orthogonal Rectangles, Filtered...... 56

Figure 35 Case 32 Orthogonal Rectangles, Larger Exponent 57

Figure 36 Case 33 Orthogonal Rectangles 58

Figure 37 Case 34 Orthogonal Rectangles, Filtered...... 59

LIST OF FIGURES (CONTINUED)

Figure 38 Case 35 Orthogonal Rectangles, Filtered, Larger Exponent 60

Figure 39 Case 36 Tank 61

Figure 40 Case 37 Tank, Targetter and Wheels Enhanced . 62

Figure 41 Case 38 Tank, Targetter and Wheels Enhanced, Smaller Exponent 63

Figure 42 Case 39 Tank 64

Figure 43 Case 40 Enahnced Tank 65

Figure 44 Case 41 Enahnced Tank, Small Exponent........ 67

Figure 45 Case 42 Enhanced Tank, Smaller Exponent 68

LIST OF TABLES

Table 4-1. Input Image Parameters 17

Table 4-2. Test Case Parameters 18

Table 4-2. Test Case Parameters (Continued) 19

ABSTRACT

This thesis analyzed a computer model of the excitation-inhibition system in a generic vertebrate retina. This model enhanced edges, eliminated brightness variations, and attenuated noise in an image. Input parameters were changed to determine their effect on the model's properties. The model was then used as a size and shape filter. Depending on the selection of input parameters, certain objects in a scene would be enhanced while others would be attenuated.

Keywords: [illegible] Retina, vision, Medicine.

I. Introduction

A. Background

In 1987, an Air Force Institute of Technology (AFIT) thesis was published entitled "A Computer Model of Inhibition, Energy Normalization, and Noise Suppression in the Retina", written by Capt Jeffery Sillart. This thesis attempted to model some of the processes which take place in all vertebrate retinas. This model reduces the effect of noise and also segment targets from the background. Test data were run through the model, and very interesting results were obtained. This model enhanced edges and was able to extract targets from noise (1). These results indicate that this model could be used for pattern recognition machines, as well as possibly increasing our understanding of the visual system.

B. Problem

The Sillart model wasn't adequately characterized, and is therefore not understood well enough to have any practical value at present. Thus, the goal of this thesis is to explore and explain some of the results of the Sillart thesis. Understanding this model is a necessary step in being able to optimize this system for different classes of targets. Without this optimization, the practical utility of this type of system for a pattern recognition machine is minimal.

C. Scope

The focus of this research will be the edge enhancement functions of the vertebrate retina. Sillart's thesis contained a model of the omatidia (individual receptor units) in the horseshoe crab eye, as well as a model of the vertebrate retina; however, the horseshoe crab eye model provided no new data which would indicate that further research is warranted. Sillart also concluded that the model he proposed might be in error. The horseshoe crab eye model was therefore not a part of this project. Similarly, the energy normalization software program written by Sillart wouldn't provide any useful data; therefore, it was not investigated.

The retina model does not model all functions of the vertebrate retina. Chapter 2 briefly describes some of the known interconnections in the retina; the computer model doesn't attempt to match this complexity. Rather, the program models one layer of connections in an attempt to study a specific retina function.

Since the results of the Sillart experiment were unexplained in his report, the original software was used. The program was not rewritten for fear of unintentionally altering the model. Once the program was understood, modifications were made to increase the efficiency and improve the ability of the model to match the actual functioning of the retina.

Different cases of the retina model were run with different input scenes. The change in output, as a function of the change in input parameters for a given scene, revealed the effect of the various model parameters.

D. Approach

The first step in this project was to obtain the original Sillart software from AFIT and verify that the original Sillart results could be duplicated.

The next step was to determine what is happening in the model. Since the model is non-linear, an attempt was made to characterize it by varying the input parameters, and seeing how these adjustments affect the results in a given input scene. The relationship between the model parameters and their effect on the scene processing could could be analyzed. This led to an understanding of the model's capability to perform edge enhancement.

The results of this phase of the research was then used to develop a technique for filtering objects of certain shapes and sizes from a scene. The parameters were set to optimize the model to pass objects of a certain size and of a certain shape. Computer-generated images were created which contained objects of the correct type, as well as objects of the incorrect type. Test cases were run to assess the model's ability to do this type of scene segmentation filtering. Tests were then run on digitized photographs of tanks, to determine how the model will work on real images.

II. Retina Functions

The retina of an animal functions as a transducer; it converts electromagnetic (light) energy into neural impulses. These impulses are then processed in the brain of the animal. However, the typical vertebrate retina also pre-processes these signals. It actually performs at least energy normalization, edge enhancement and noise supression(2). In human beings, these functions sharpen the image, and allow the person to look at a bright corner of a room and a dark corner, and see equally well. In the case of lower vertebrates, the various retinal preprocessing functions enable the animal to make use of the electro-magnetic energy to detect food, sense danger, etc., without having an extensive cerebrum.

The vertebrate retina is made up of a layer of photoreceptors and several layers and types of neurons, the last layer of which drives the optic nerve and transmits the preprocessed data to the brain. The connections between the retinal cells, the relative weighting of their interconnections, and the specific cell functions enable the retina to perform the inent preprocessing functions. The various types of retinal cells are termed photoreceptors, horizontal cells, bipolar cells, amacrine cells, and ganglion cells (5:71-72).

Figure 1 illustrates how these different cells types are connected. Light energy enters the eyeball and is

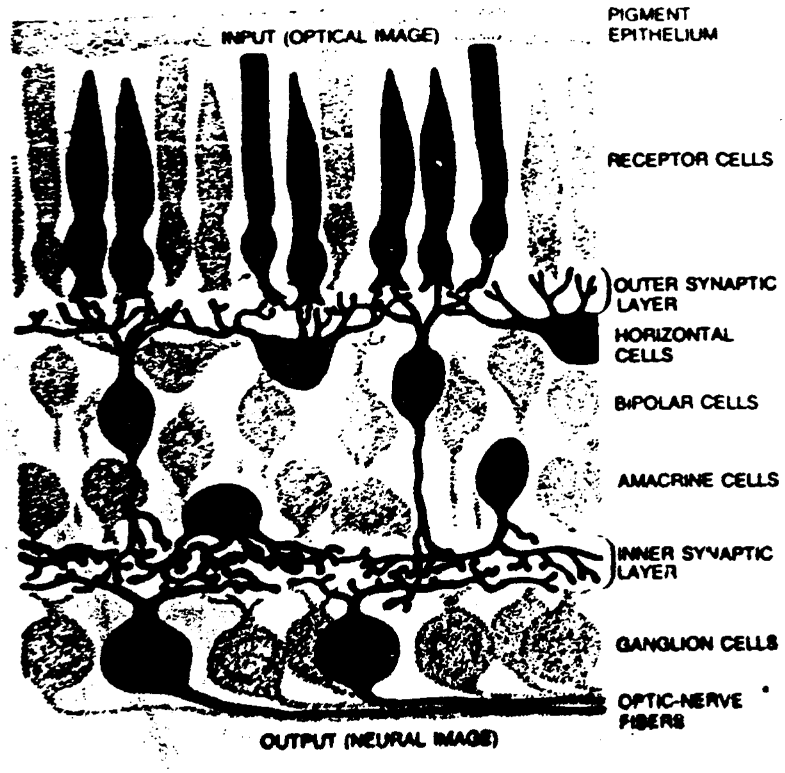


Figure 1 Retinal Cell Connections (5:75)

focused onto the photoreceptors. These cells, often refered to as rods and cones, convert the incoming photons into neural electro-chemical signals. The output of these cells is a release of a neurotransmitter chemical into the synaptic junction between them and the bipolar cells, which in turn signal the ganglion cells. The output of the ganglion cells is a stream of 100 mV impulses transmitted through the optic nerve to the brain (2). These neural layers provide the data path from the photoreceptors to the optic nerve.

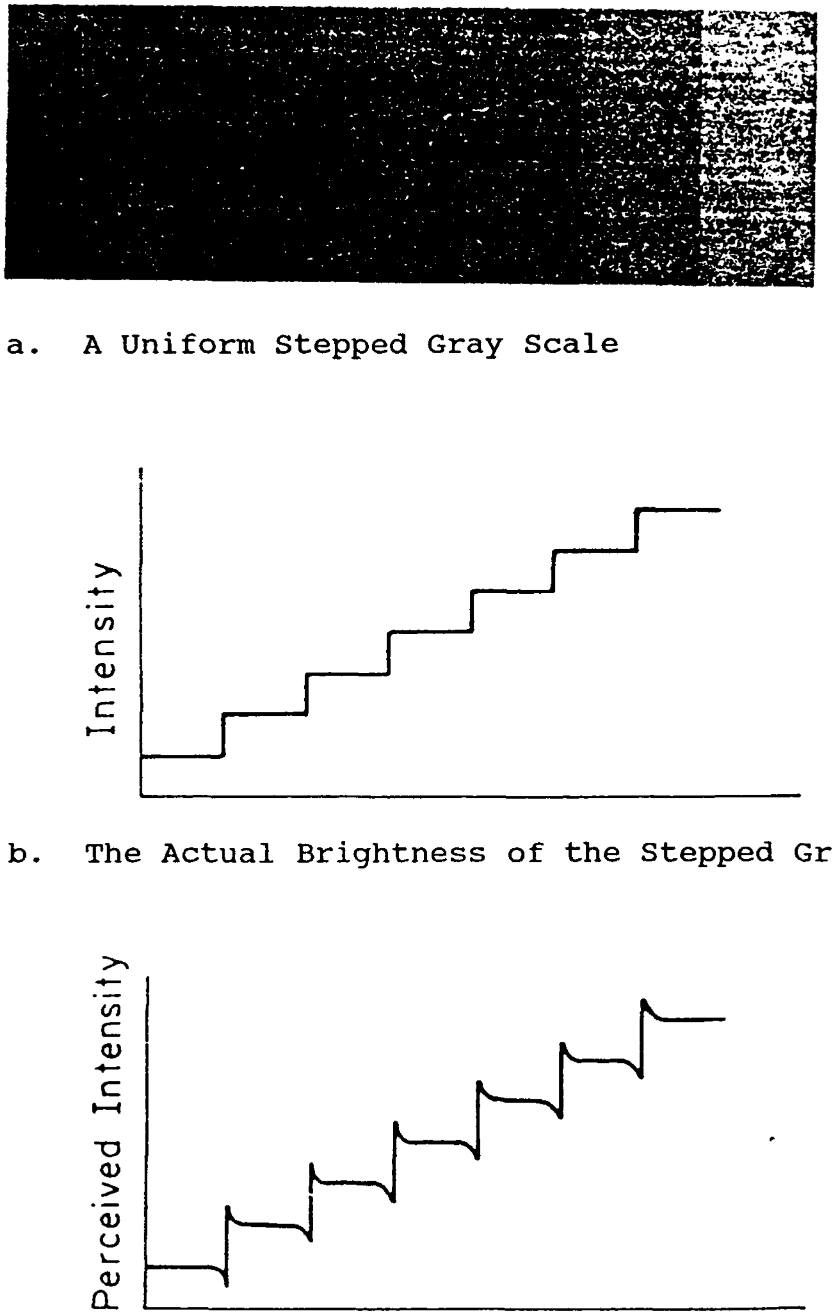
In addition to this throughput pathway, information is also carried laterally, across the retina. The neurotransmitter released from the photoreceptor also triggers the horizontal cell. These cells in turn can release neurotransmitters which can affect photoreceptors, other horizontal cells, and bipolar cells. Likewise, amacrine cells can affect bipolar cells, other amacrine cells, and ganglion cells (4:73).

These lateral connections enable the retina to perform some of its preprocessing functions. The horizontal and amacrine cells allow the signal being transmitted through one photoreceptor to optic nerve pathway to be influenced by the signal being transmitted through the neighboring pathway.

This two-dimensional, complex, non-linear, multiple feedback loop system enables the retina to perform many different functions. The operating range of the photoreceptors adjusts itself based on the average ambient intensity level (5:70-72). In addition, the horizontal cells carry information which shifts the operating range of the bipolar cells, which allows maximum contrast at different levels of photoreceptor output. Also, the amacrine cells carry information about changes in the local area which affects the response of the retina to a change in intensity.

The vertebrate retina can also perform edge sharpening and contrast enhancement because of an excitation and an inhibition network provided by the lateral connections. When a particular photorecetor region is illuminated, the lateral connections cause the local area around that particular receptor to be more easily excited consequent to the impinging photons. This local area is termed the excitation region. The photoreceptors more distant from that particular photoreceptor, however, are inhibited. Thus, for instance, retinal areas which would normally be triggered by the blurry edge of an object are prevented from firing. These photoreceptors are said to be in the inhibition region (2).

An illustration of this effect is shown in Figure 2. Figure 2a shows a picture of a series of vertical bars of increasing intensity. The actual intensity of this pattern, is shown in Figure 2b. Due to the lateral connections in the retina, however, the reader will see the pattern as shown in Figure 2c.



a. A Uniform Stepped Gray Scale

b. The Actual Brightness of the Stepped Gray Scale

c. The Subjective Perceived Intensity of the Gray Scale

Figure 2 Illustration of Lateral Connection Effects (1:199)

While the retina performs many different functions in the vertebrate visual system, edge enhancement and shape filtering are the subjects of this research project. A computer model derived from Sillart's research was used to model this function, and test cases were run under different input conditions. This model is described in the next chapter.

III. Experimental Design

A. Equations for the Output of Retina Cells

The potential across the membrane of a typical vertebrate retinal cell when both the excitation and inhibition regions are fully illuminated by a given input scene was calculated by the computer program. The formula fo:· this potential was developed by a curve fit to experimental data from in vitro measurements of the mudpuppy salamander (Necturus Maculosus) retina (4:63 and 67). As implemented by Sillart (3:3-4), the formula is:

$$V = (V_{max} * I^n)/(I^n + K^n) \qquad (3-1)$$

where:

V = output of a retina cell
V_{max} = maximum retina cell output
I = average intensity on the excitation region
K = average intensity on the inhibition region
n = measure of steepness of inhibition curve
n between 0.7 and 1.0 for receptor cells
n between 1.4 and 3.0 for bipolar cells
n between 3.0 and 4.0 for ganglion cells

This formula can be rewritten as

$$V/V_{max} = 1/(1 + X^n) \qquad (3-2)$$

where $X = K/I$

A plot of this equation is shown in Figure 3. To understand how the edge enhancement occurs, consider the boundary between a low intensity region and a high intensity region. Further, take each pixel in the image to be a distinct retinal cell. Then, if the retinal cell of interest is far away from the boundary, i.e. the inhibition region doesn't

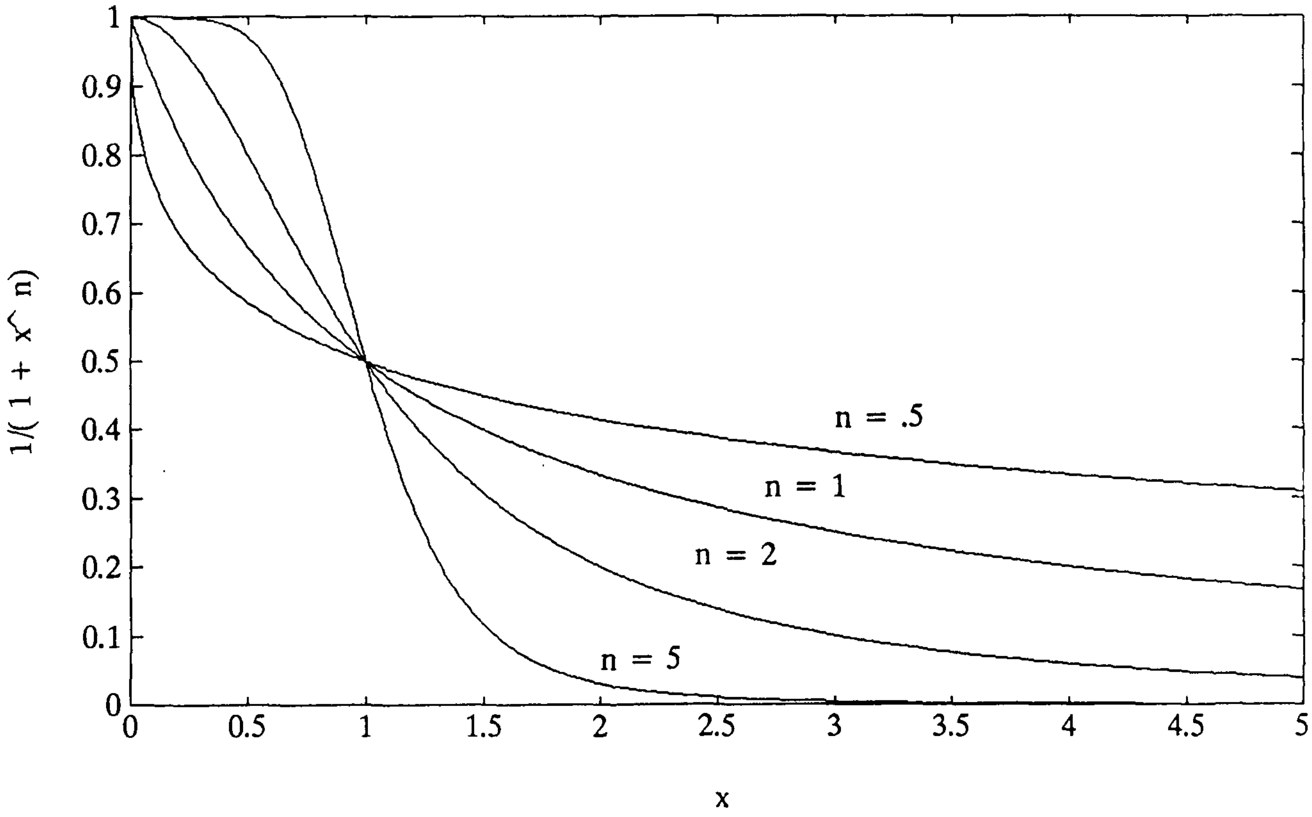


Figure 3 Plot of Equation 3-2

cross the boundary, all cells in the excitation region and inhibition region are stimulated at the same intensity. The ratio K/I is unity, therefore the output of that specific cell is half the maximum value.

However, for pixels close to the transition border on the low intensity side, many of the pixels in the inhibition region will be in the high intensity region. Thus, the value K/I is increased, and V is consequently reduced below one half of the maximum value. Closer to the boundary, still more pixels of the inhibition region are in the excitation region, and therefore the ratio K/I is further increased, so V is reduced even further.

For retina cells near the border on the high intensity side, many cells in the inhibition region are at the low intensity side. The ratio K/I is now less that unity, therefore the output of the cell is increased above one half maximum. The further away from the border, the less pronounced the increase in output, since less cells of the inhibition region are in the low intensity side.

For pixels far away from the boundary, on the high intensity, side, the ratio K/I is unity. Thus, the output of the cell is half the maximum value.

B. Input Variables

The program begins by asking the user to input a number of parameters. These parameters determine the image to be used and the size and strength of the lateral inhibition and excitation.

First, the user is asked to input a '0' for images in the data base or '1' for camera images. For this experiment, camera images weren't used, so '0' was always input. This parameter was left as a variable so future research could be done using this program on real images.

The next input variable is the blocksize. Sillart put a routine in his program which calculated the average for the pixels in a blocksize by blocksize section, and replaced all of the pixels values in that region by the average. This may have come from an attempt by Sillart to model the omatidia in the eye of the horseshoe crab. Since performing this averaging would only blur the results, blocksize was always set to '1'.

The program then asks for the size of the excitation region. The first parameter is the number of pixels to the right and left of the pixel of interest within the excitation region. The second parameter is the number of pixels above and below the pixel of interest within the excitation region.

The next set of input parameters determines the size of the inhibition region. The user inputs the number of times the inhibition region is wider than the excitation region, and the the number of times the inhibition region is higher than the excitation region.

The user is then asked to input the size of the region over which the signal is to be averaged. No purpose for this routine could be determined, so this value was always set to '0'.

The next variable to be input is the exponent 'n' from Equation 3-1. A wide range of values was input to determine the effect of this exponent on the edge enhancement capability of the retina model.

The user then inputs a filename. This allows the program to store each test case in a separate output file.

The final input parameter determines which of the images in the data base will be used. Possible options were a noise-free rectangle, three rectangles in noise, and a series of vertical bars, similar to those shown in Figure 1. A fourth option in the program was unavailable for this experiment.

Chapter IV discusses the choice of input parameters for this experiment, and shows the effects that the choice of input values has on the output.

C. Program Implementation

The program first calculates the excitation region. It does a test to determine if the size of the excitation region would place the region outside the image. If so, the region is truncated. In any case, the upper, lower, rightmost and leftmost coordinates of the excitation region are calculated.

The program then calculates the coordinates on the inhibition region. Again, a test is performed to ensure the inhibition region doesn't go off the image. The upper, lower, right most and left most coordinates of the inhibitor region are calculated.

The intensity values of the pixels in the excitation region around the pixel of interest are summed, and the number of pixels around the pixel of interest are calculated. Also, the intensity values in the inhibition region are summed, and the number of inhibitory pixels are counted. The program then calculates the average values of the excitation and inhibition regions.

Once this has been completed, the program calculates the result of formula 3-1, using 255 as the maximum output. This yields the value of the current pixel of interest in the output image. The process is repeated for all pixels in the output image. Then, depending on user input, the program will repeat or terminate.

A listing of the program is given in the appendix.

IV. Results and Analysis

A. Introduction

This chapter describes the results of the retina computer simulations. The experiment can be thought of as consisting of two phases. Phase I was an investigation into the edge enhancement properties of the model. Phase II applied the results of the previous phase to study the feasibility of developing of a filter which could be tuned to attenuate the intensity levels of objects which are of no interest to the pattern recognition system, based on size and shape.

Each image shown has a horizontal line running through it. The graph at the bottom of each image shows the gray scale value of each pixel along the line. This was done to facilitate the analysis. Black (gray level 0) is at the bottom of the graph, and white (gray level 255) is at the top.

Table 4-1 describes the images used as input for the test cases. Table 4-2 shows the values used for the model parameters for each test case discussed in the following sections.

B. Edge Enhancement.

Figure 4 shows a noise-free rectangle which was input to the retina model. Figures 5 through 7 show the effect of constant-sized excitation and inhibition regions, with

TABLE 4-1 INPUT SCENE PARAMETERS

CASE 1.	Rectangle 144 x 144 pixels, gray level 25 Background Gray level 55
CASE 9.	Vertical bars 64 pixels wide, gray levels 64,80,96,112,128,144,150,176
CASE 15.	Squares, gray level 3,5,7 (left to right) prior to adding noise
CASE 26.	Squares, 145 x 145, 45 x 45, 9 x 9 pixels gray level 25 Background gray level 55
CASE 30.	Rectangles, 31 x 11, 11 x 31, gray level 25 Background level 55
CASE 33.	Rectangles, 31 x 11, 11 x 31, gray level 85 Background level 55
CASE 36.	N/A
CASE 39.	N/A

TABLE 4-2 TEST CASE PARAMETERS

	WIDTH	HEIGHT	WIDTH	HEIGHT	
CASE	EXCIT	EXCIT	INHIB	INHIB	EXPONENT
#	REGION	REGION	REGION	REGION	
1	N/A	N/A	N/A	N/A	N/A
2	3	3	23	23	5
3	3	3	23	23	10
4	3	3	23	23	15
5	3	3	33	33	10
6	3	3	5	5	10
7	11	11	31	31	10
8	3	3	23	23	10
9	N/A	N/A	N/A	N/A	N/A
10	3	3	23	23	5
11	3	3	23	23	10
12	3	3	23	23	15
13	3	3	33	33	10
14	3	3	5	5	10
15	N/A	N/A	N/A	N/A	N/A
16	3	3	23	23	5
17	3	3	23	23	10
18	3	3	23	23	15
19	3	3	33	33	10
20	3	3	43	43	10
21	3	3	5	5	10
22	21	21	41	41	10

TABLE 4-1 TEST CASE PARAMETERS (CONTINUED)

CASE	WIDTH EXCIT REGION	HEIGHT EXCIT REGION	WIDTH INHIB REGION	HEIGHT INHIB REGION	EXPONENT
23	21	21	101	101	10
24	3	3	33	33	15
25	3	3	43	43	15
26	N/A	N/A	N/A	N/A	N/A
27	41	41	43	43	2
28	41	41	81	81	2
29	41	41	43	43	1
30	41	41	43	43	1
31	N/A	N/A	N/A	N/A	N/A
32	11	31	13	33	1
33	N/A	N/A	N/A	N/A	N/A
34	11	31	13	33	1
35	11	31	13	33	2
36	N/A	N/A	N/A	N/A	N/A
37	13	15	17	19	10
38	13	15	17	19	5
39	N/A	N/A	N/A	N/A	N/A
40	21	17	25	21	10
41	21	17	25	21	5
42	21	17	25	21	1

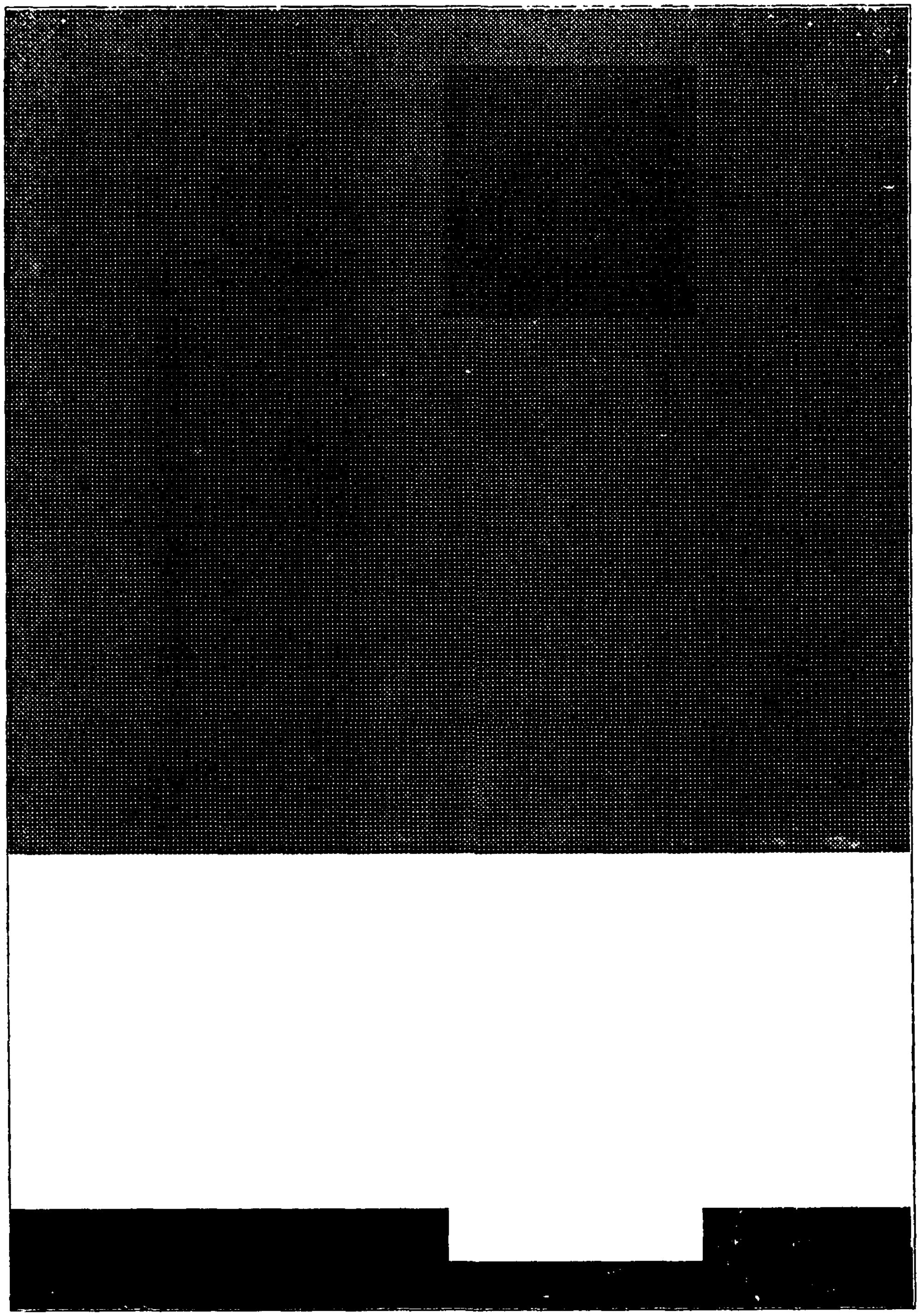


Figure 4 Case 1. Noise-free Rectangle

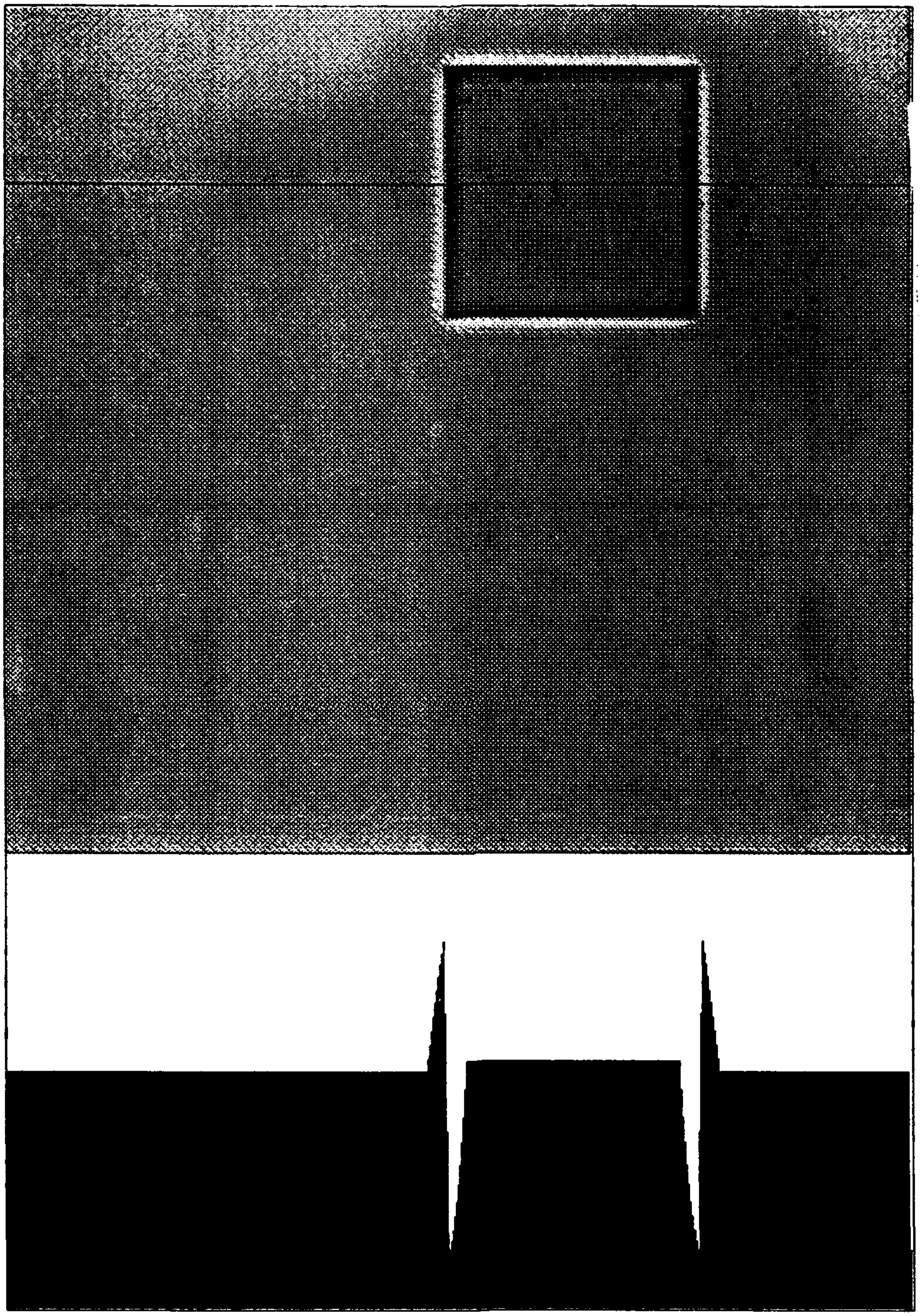


Figure 5 Case 2. Edge-enhanced Noise-free Rectangle

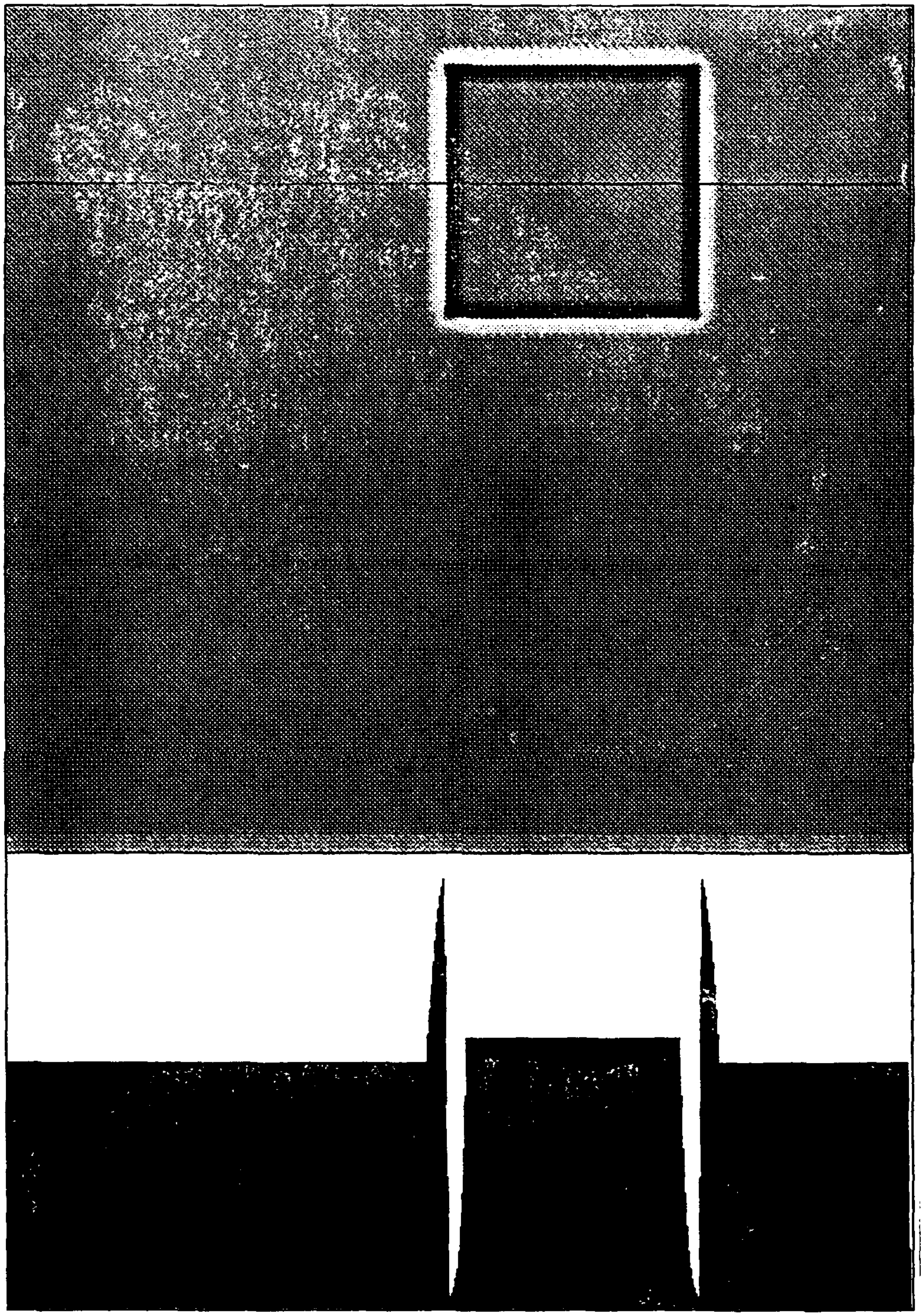


Figure 6 Case 3. Edge-enhanced Noise-free Rectangle, Increased Exponent

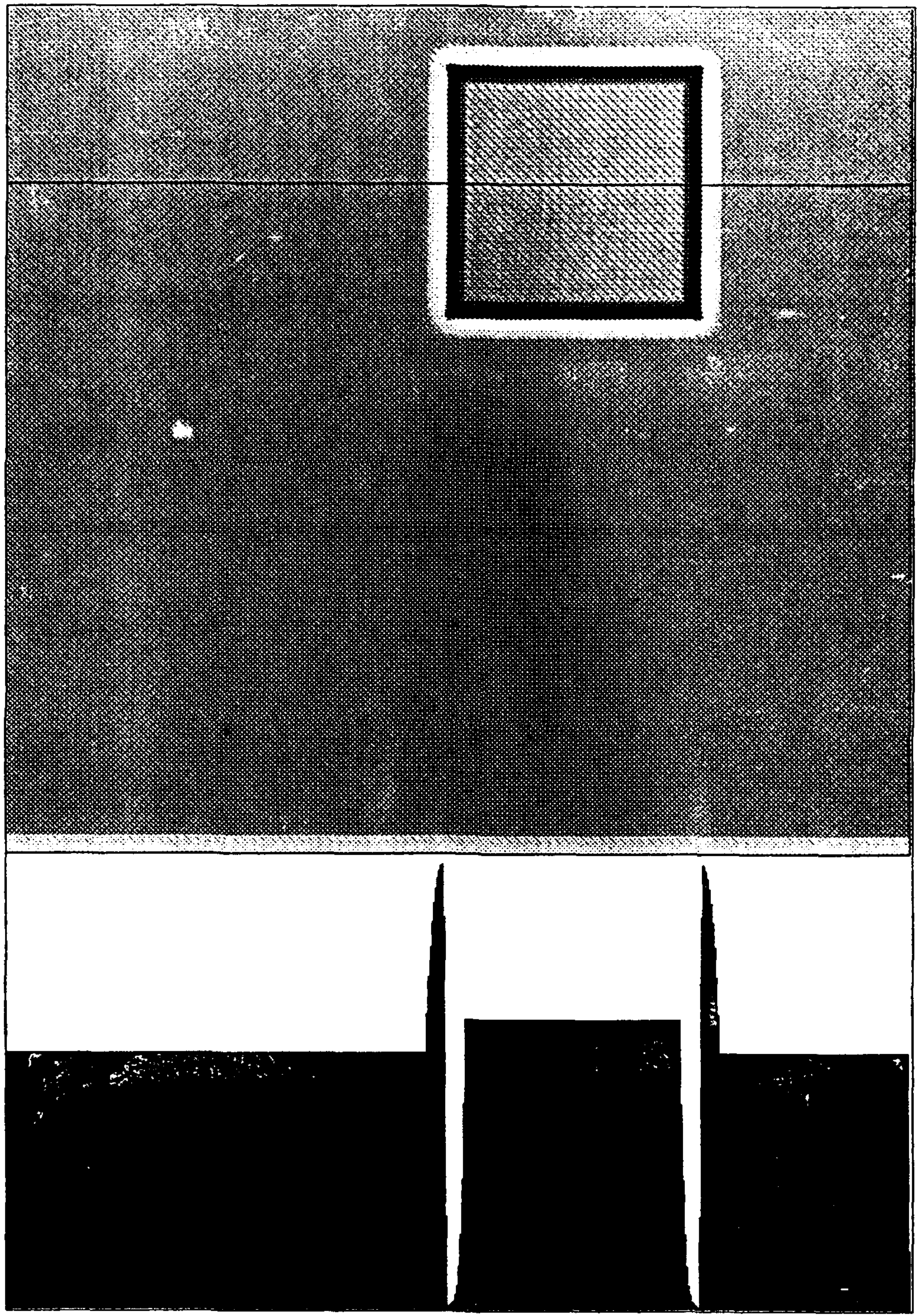


Figure 7 Case 4. Edge-enhanced Noise-free Rectangle, Increased Exponent

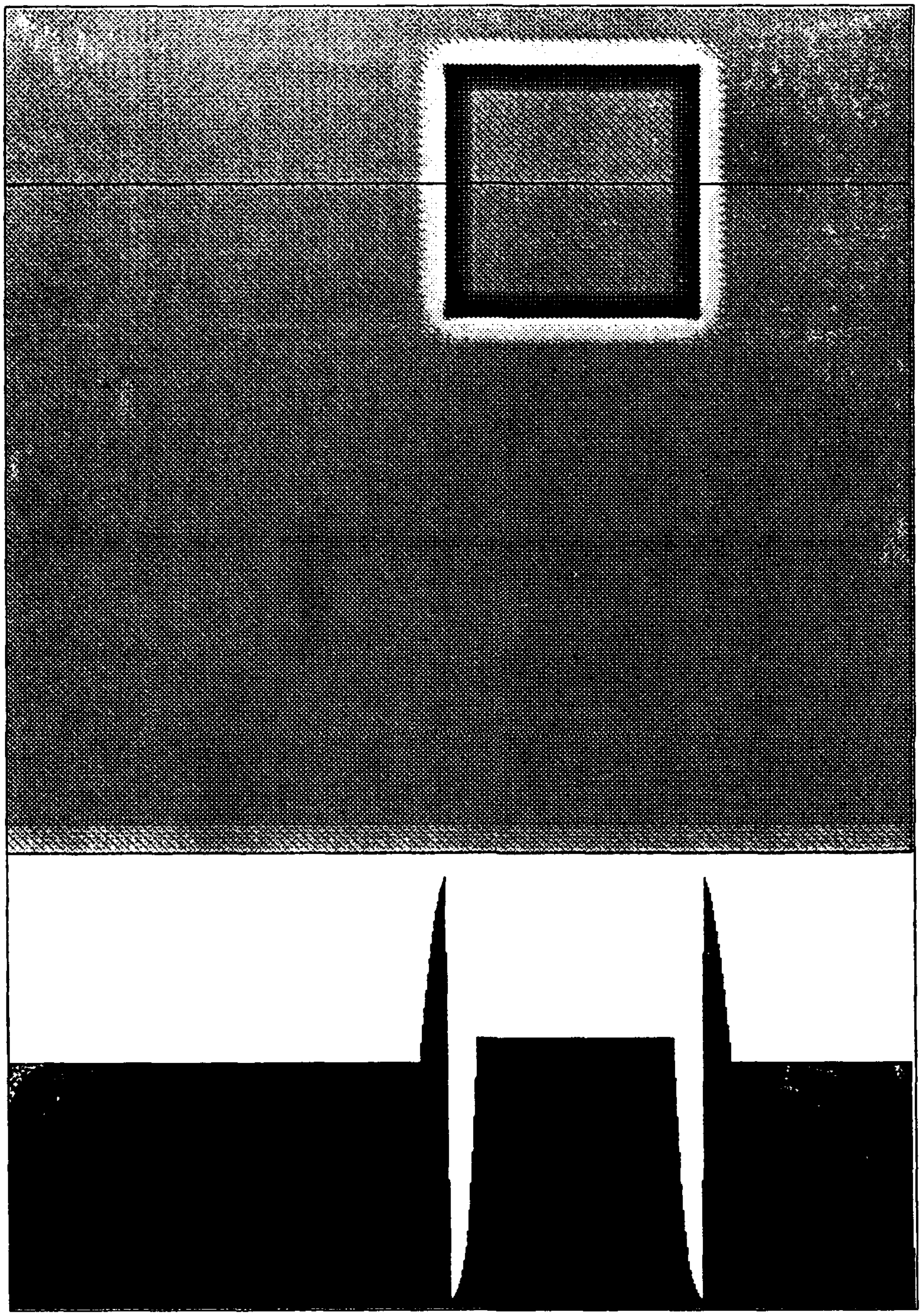


Figure 8 Case 5. Edge-enhanced, Noise-free Rectantle, Larger Inhibition Region

increasing value of exponent. Figure 8 shows the effect of increasing the size of the inhibition region, Figure 9 shows the effect of increasing the size of both the excitation and inhibition regions. Figure 10 shows the effect of decreasing the size of the inhibition region.

As the exponent is increased, with all other parameters being held constant, the magnitude of the peaks and valleys occurring in the processed image wherever the original image had transients increases. Increasing the size of the inhibition region increased the width of the peaks and valleys because pixels further from the transition are affected by the transition.

These results are consistent with the analysis presented in Chapter III. Increasing the value of the exponent magnifies the ratio of the average intensity on the inhibition region and the average intensity on the excitation region, thereby increasing the height of the peaks and the depths of the valleys. Increasing the size of the inhibition region increases the thickness of the peaks and valleys, since pixels further from the transition will be affected by the transition. Decreasing the size of the inhibition region decreases the thickness of the peaks and valleys, since only those pixels very close to the transition will be affected.

However, a result seemingly inconsistent with the analysis in Chapter III appears in these figures. The gray

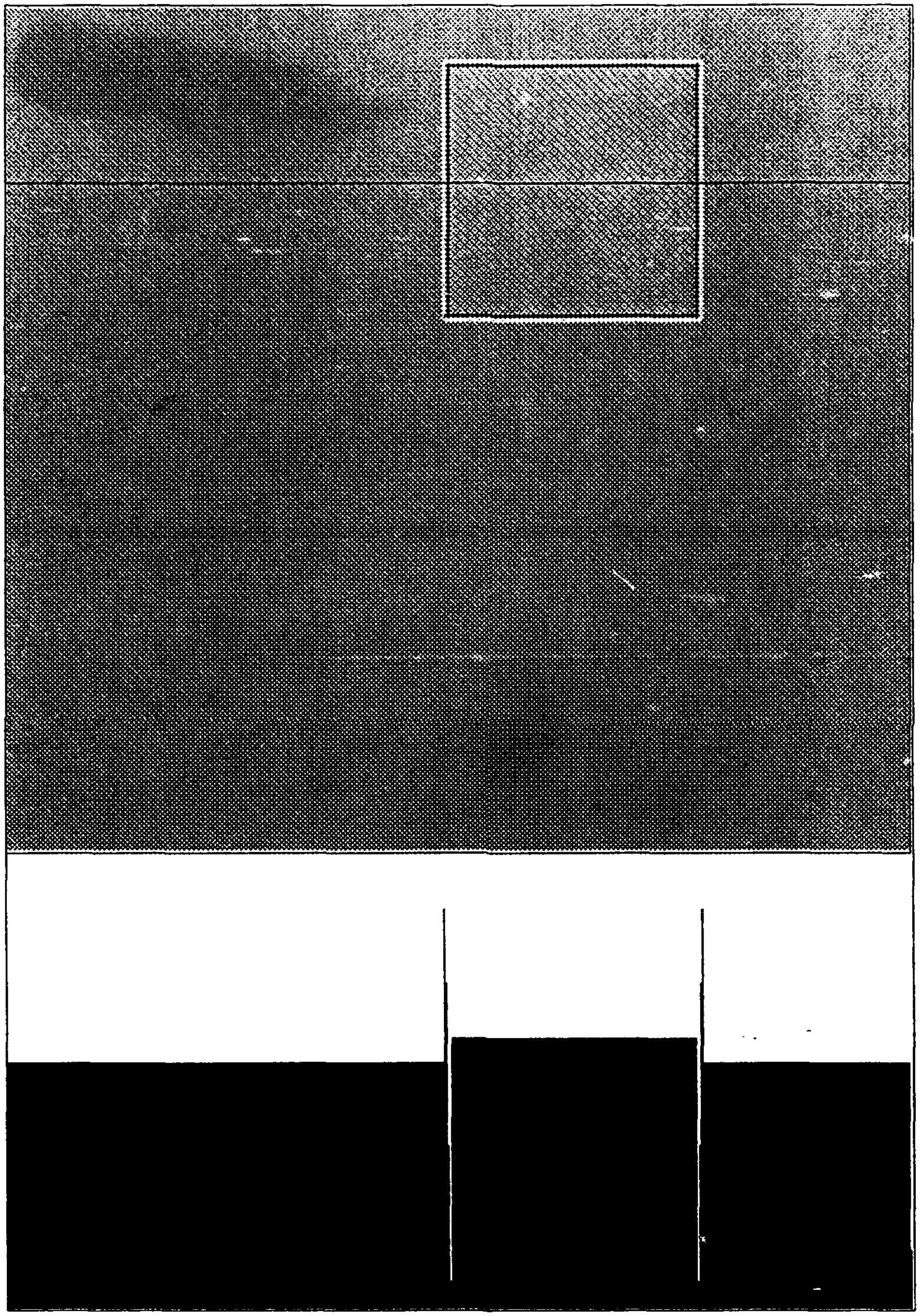


Figure 9 Case 6. Edge-enhanced Noise-free Rectangle, Larger Excitation and Inhibition Regions

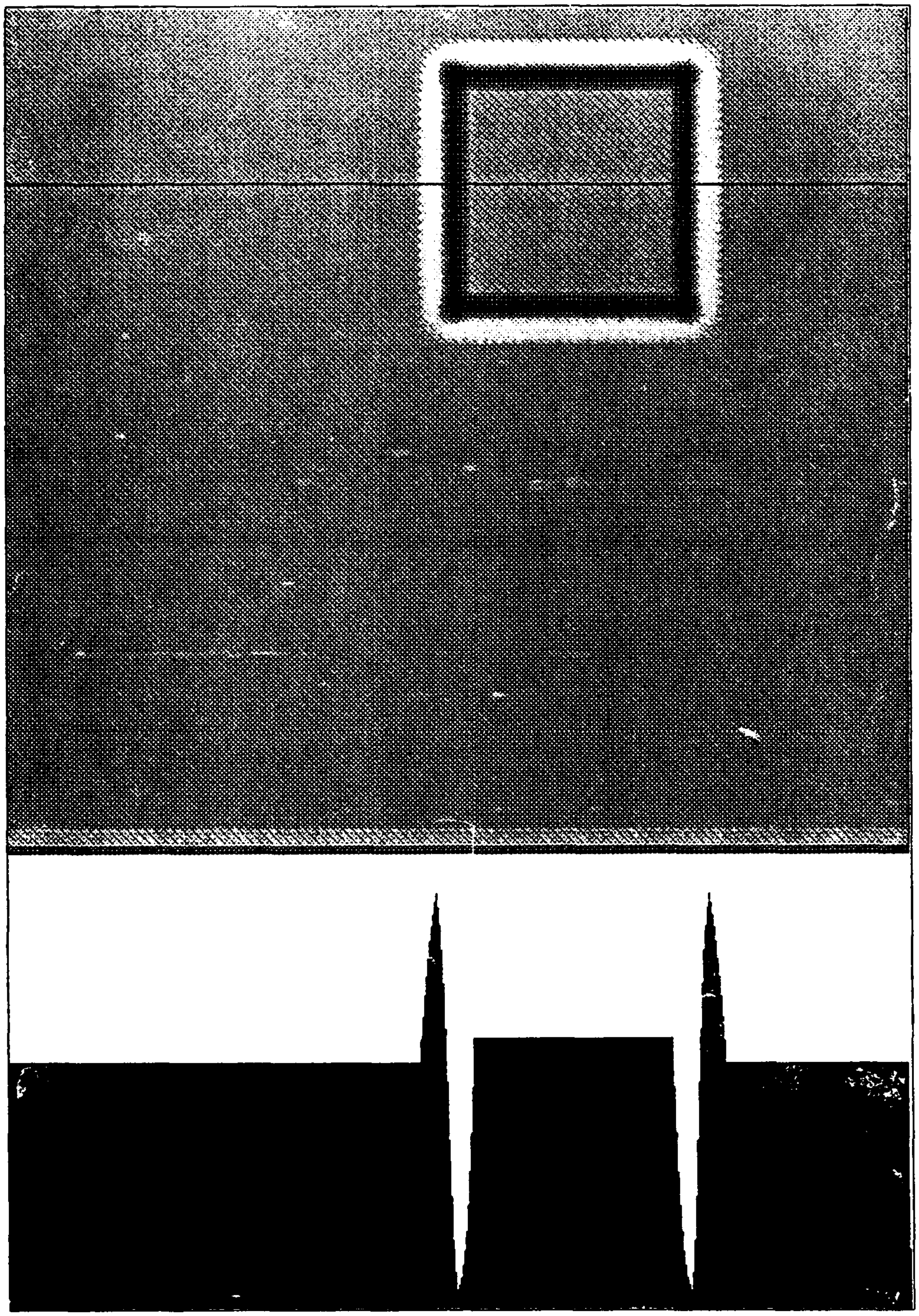


Figure 10 Case 7. Edge-enhanced Noise-free Rectangle, Smaler Inhibition Region

level of the rectangles and the background is not set at one-half the maximum, or 128 out of a possible 256 levels, as the analysis predicts. In fact, the background and the rectangle are both at gray levels above 128, with the rectangle being at a slightly higher intensity. The difference in intensity increases as the value of the exponent increases. This happens because the Sillart program increments the average intensity of the excitation region by one before calculating Formula 3-1. The values of the average intensity on the excitation region and the average intensity on the inhibition region are no longer equal. Further, since the rectangle was originally at a lower intensity than the background (gray level 25 vs. gray level 55), the addition of '1' to the excitation region average value has a more pronounced effect at the rectangle than at the background (56/55 vs. 26/25). This addition is necessary to prevent the program from prematurely terminating under certain input conditions, due to a problem with the exponentiation procedure.

To show the effect of incrementing the average of the excitation region, a test case was run with the increment statement removed. The result is shown in Figure 11. The value of all points away from the transition regions, both in the rectangle and the background, are at gray level 128, as expected.

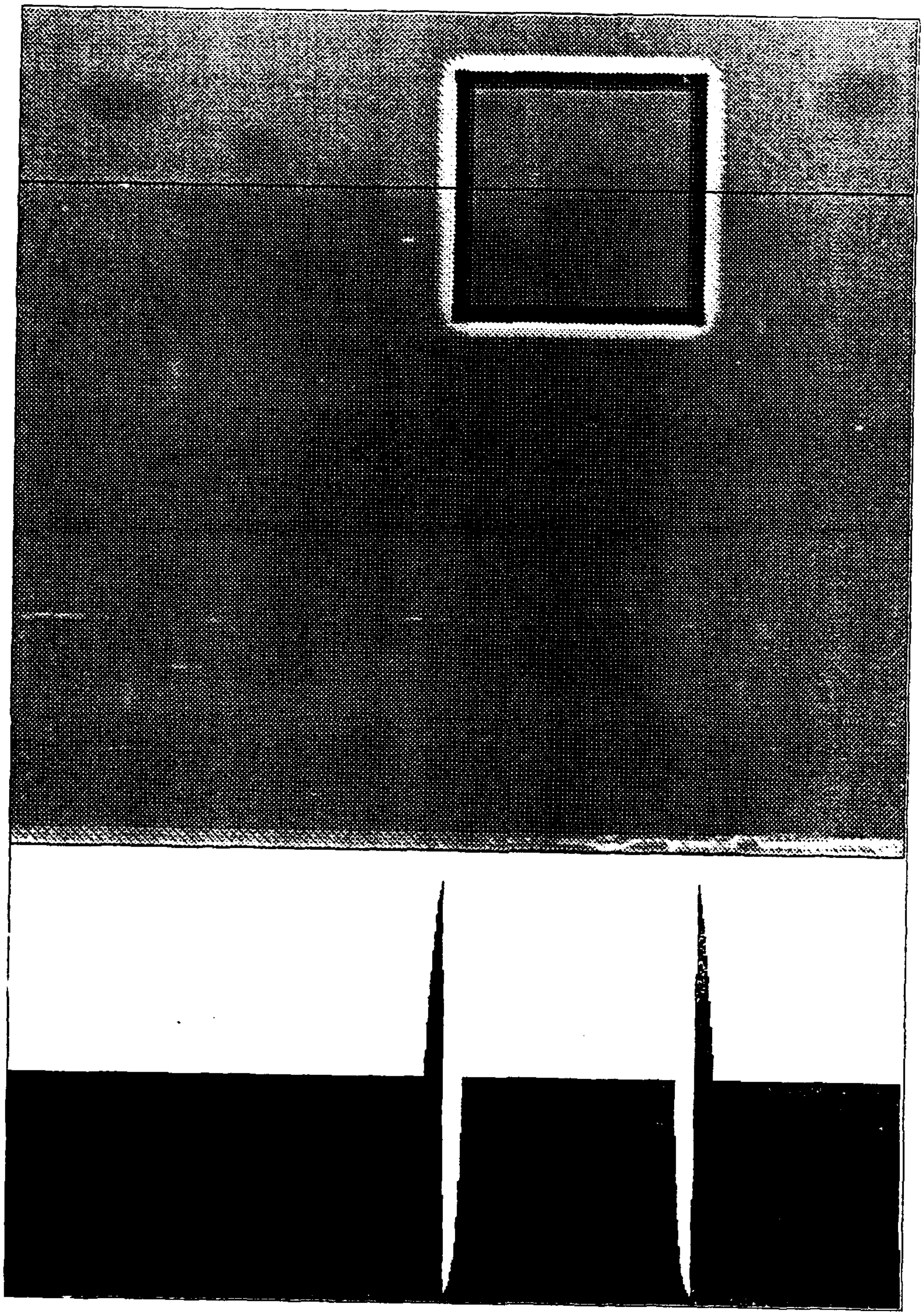


Figure 11 Case 8. Edge-enhanced Rectangle, Without Additional Increment

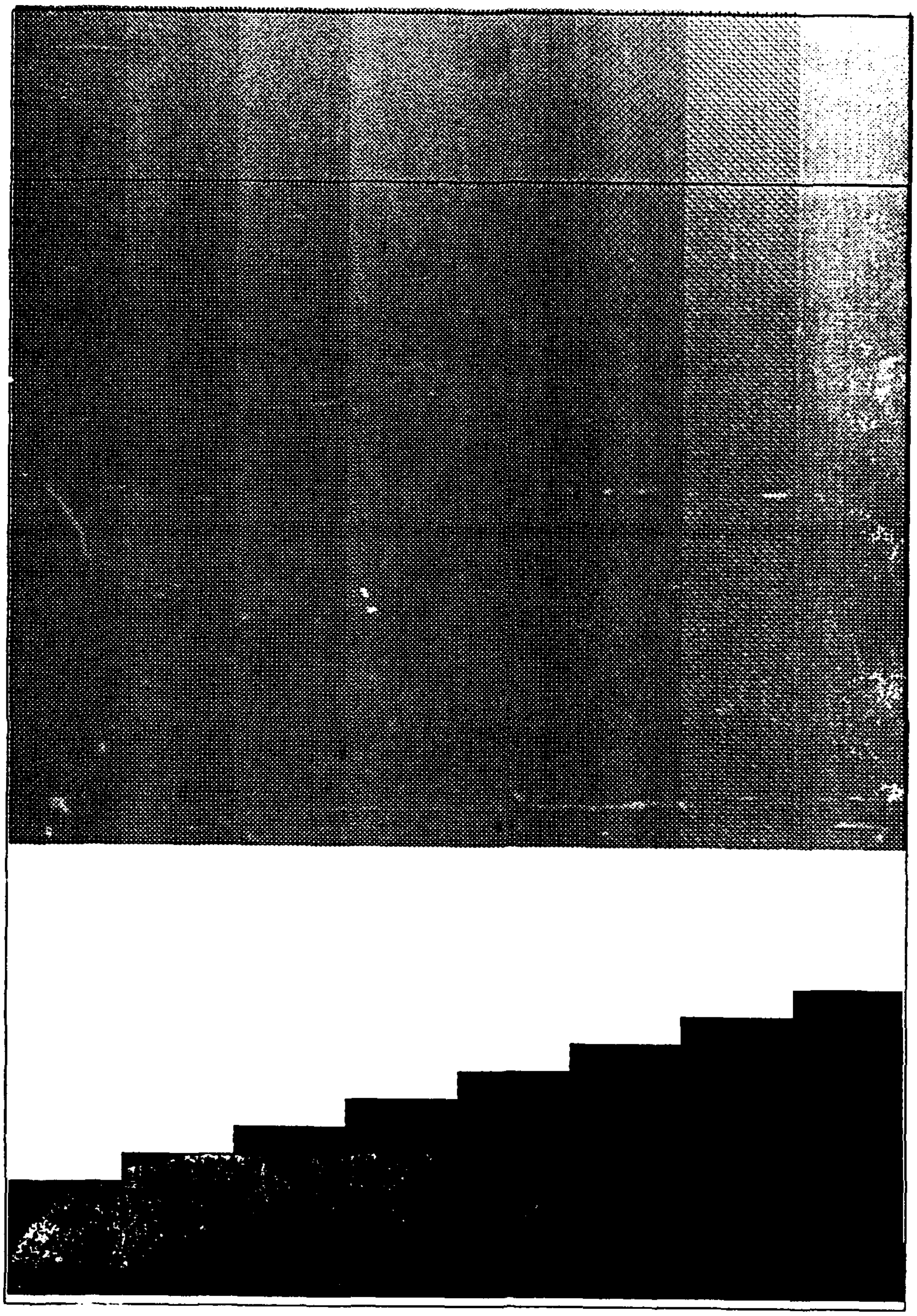


Figure 12 Case 9. Vertical Bars

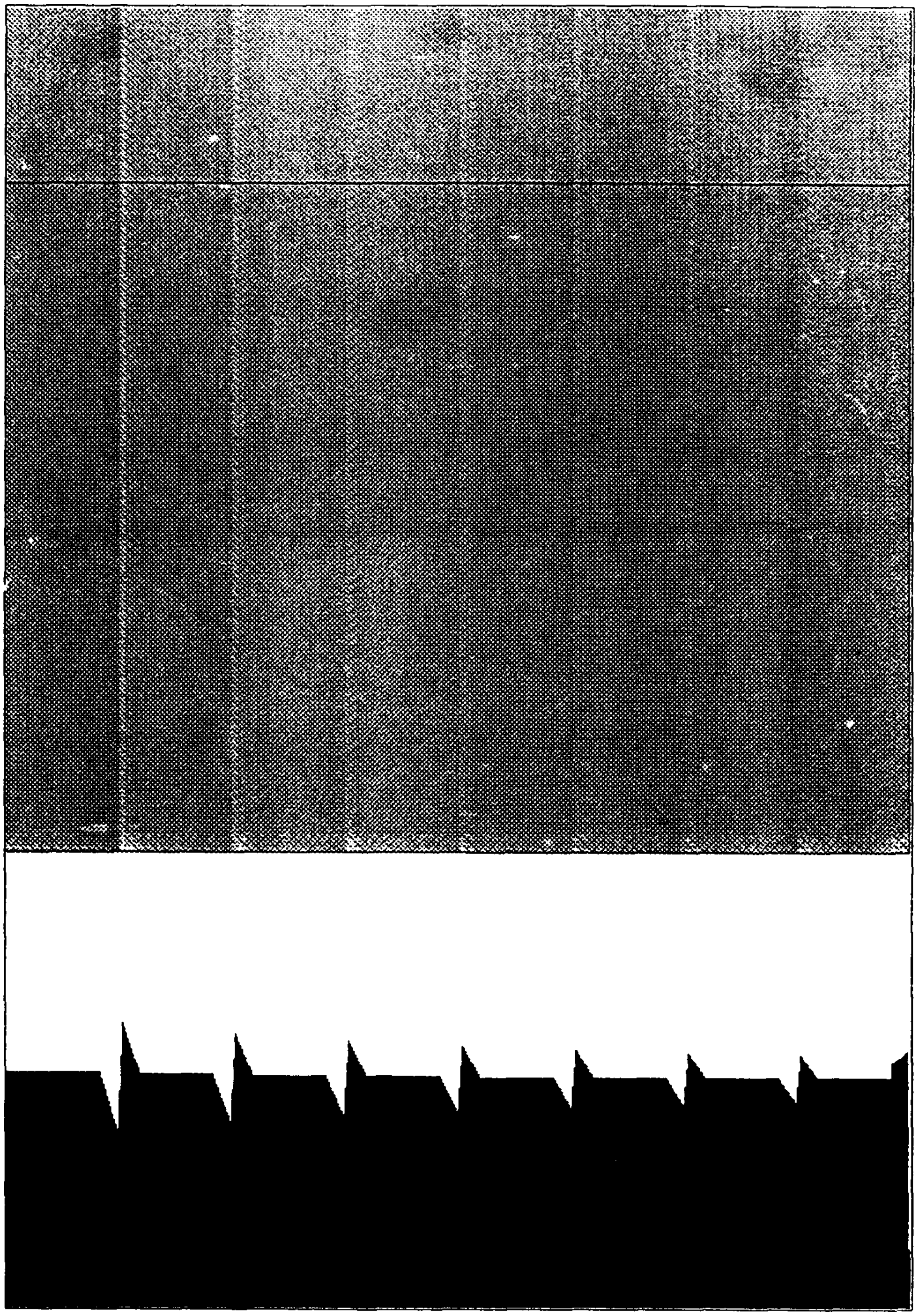


Figure 13 Case 10. Edge-enhanced Vertical Bars

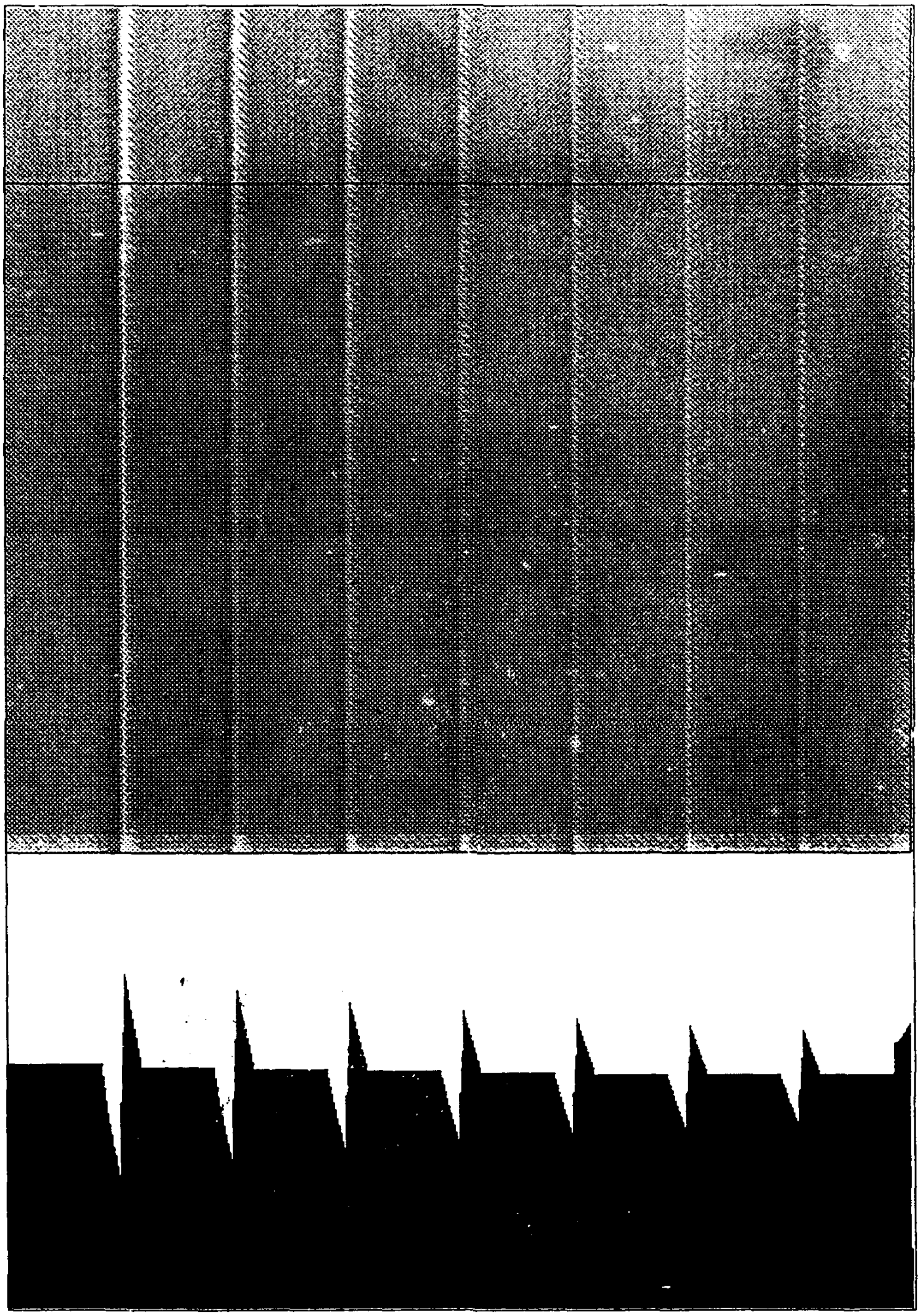


Figure 14 Case 11. Edge-enhanced Vertical Bars, Increased Exponent

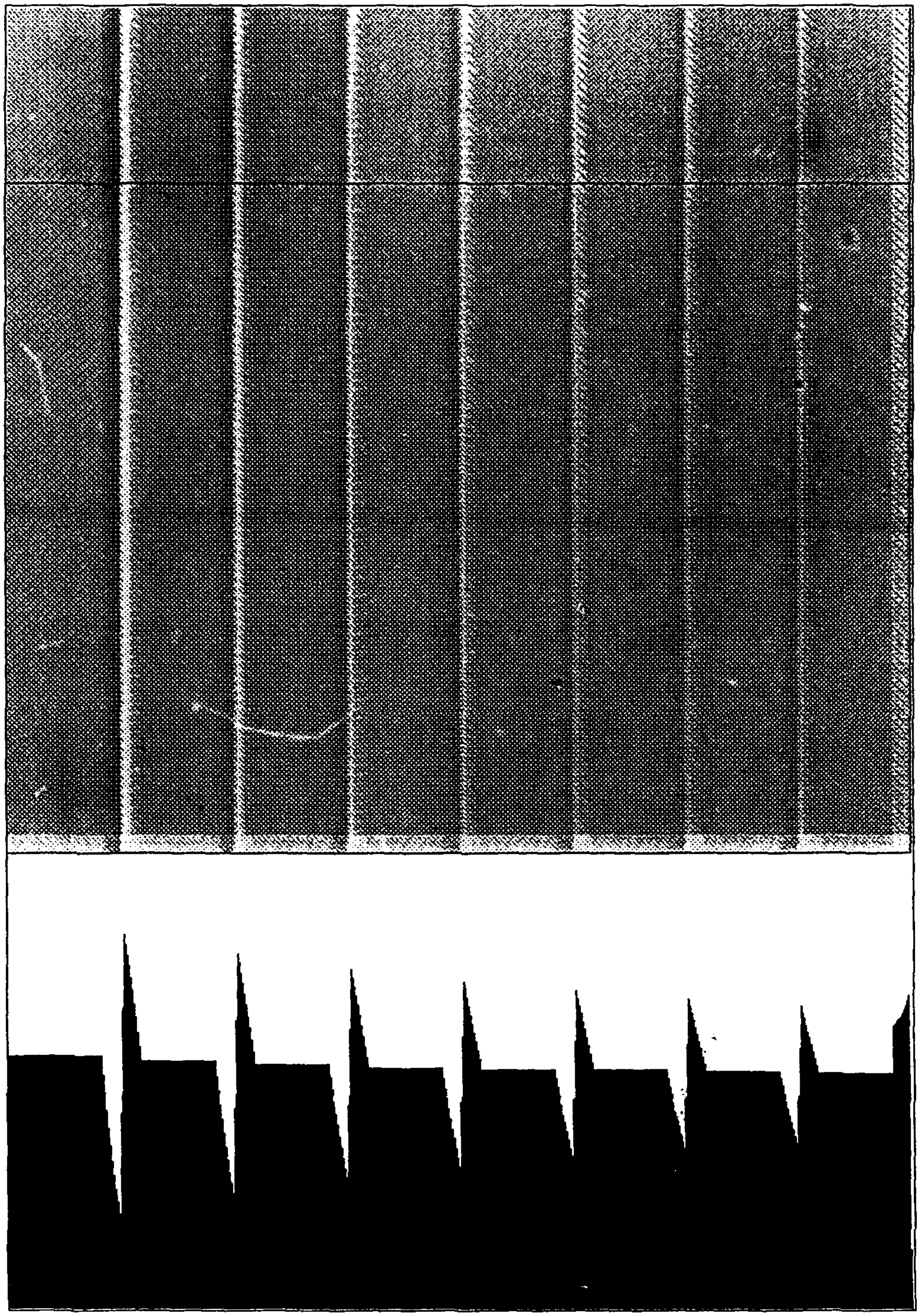


Figure 15 Case 12. Edge-enhanced Vertical Bars Increased Exponent

Figure 12 shows a series of vertical bars of constant width and increasing intensity. Figures 13 through 15 shows the result of increasing the exponent while keeping the excitation and inhibition regions constant. The results are completely consistent with the results of Figures 5 through 9. Note the similarity to the edge enhancement pattern shown in Figure 5. The magnitude of the peaks and valleys decreases from left to right, with increasing intensity level. This is because the degree of enhancement is a function of the ratio of average intensities. The bars increase in intensity from left to right at a constant rate, so the percentage change of that increase is greater at lower intensity levels.

Figures 16 and 17 show the results of increasing and decreasing the size of the inhibition region, respectively. Again, there is complete consistency with the results of the same tests on the rectangle.

Figure 18 shows three low-intensity rectangles against a noisy background. These rectangles are 1 inch wide, and are centered at 1 inch, 1.5 inches, and 4 inches from the left edge of the figure. These rectangles are at gray level 7, 5, and 3, respectively. The background is actual digitized video from a camera looking at a uniform white background and shows the typical residual noise in video systems. Figures 19 through 21 show the result of increasing the exponent while keeping the size of the

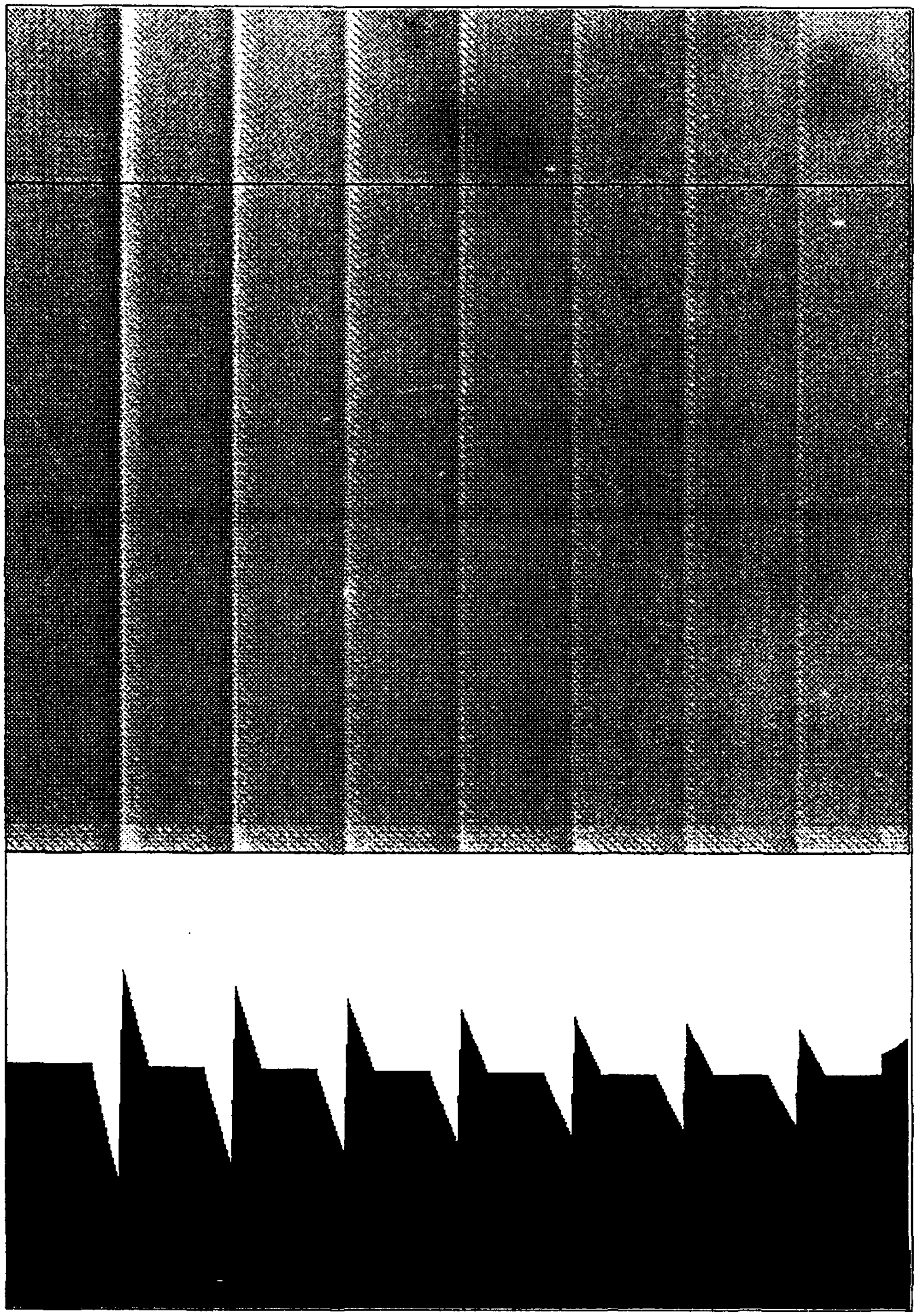


Figure 16 Case 13. Edge-enhanced Vertical Bars, Larger Inhibition Region

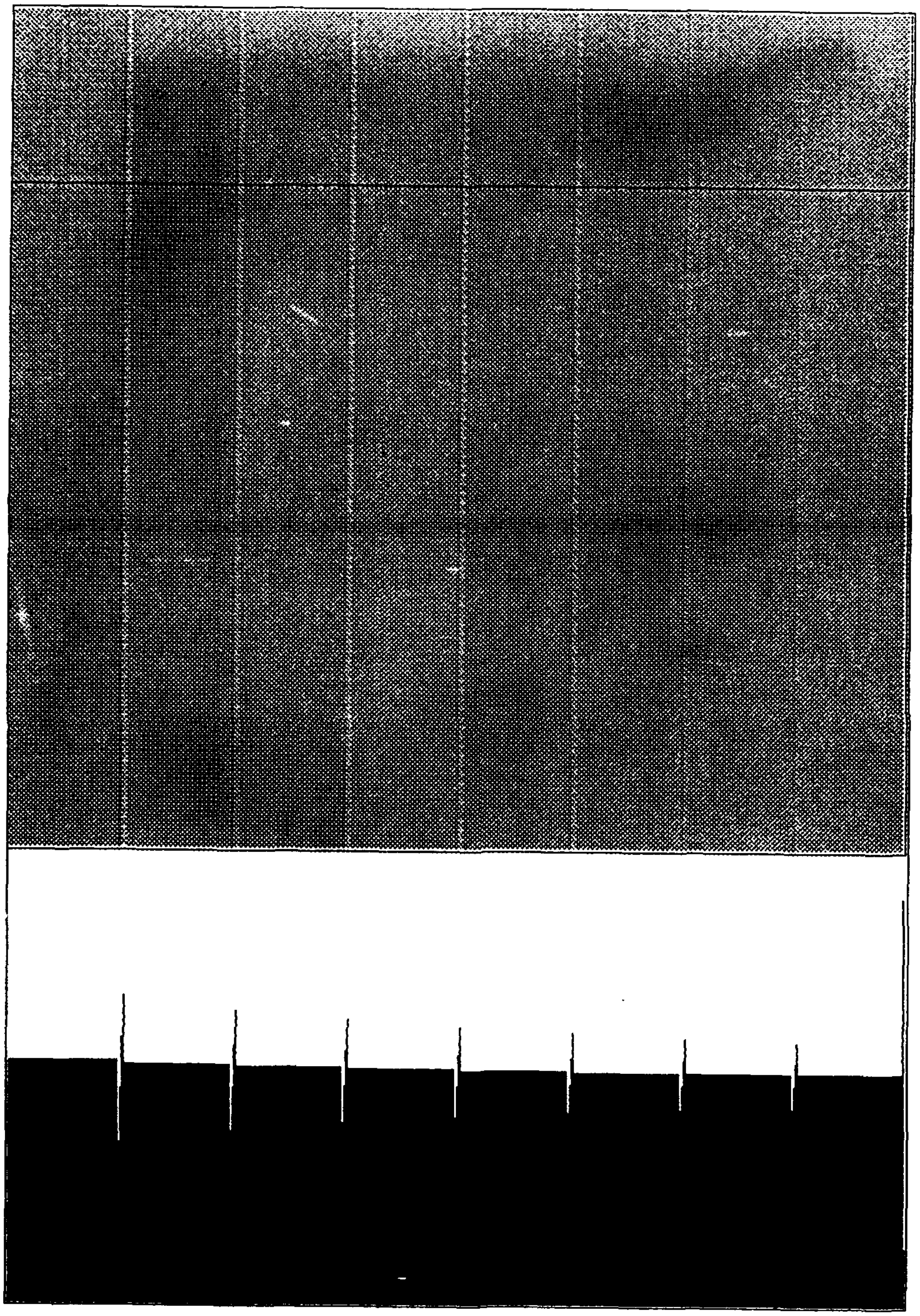


Figure 17 Case 14. Edge-enhanced Vertical Bars, Smaller Inhibition Region

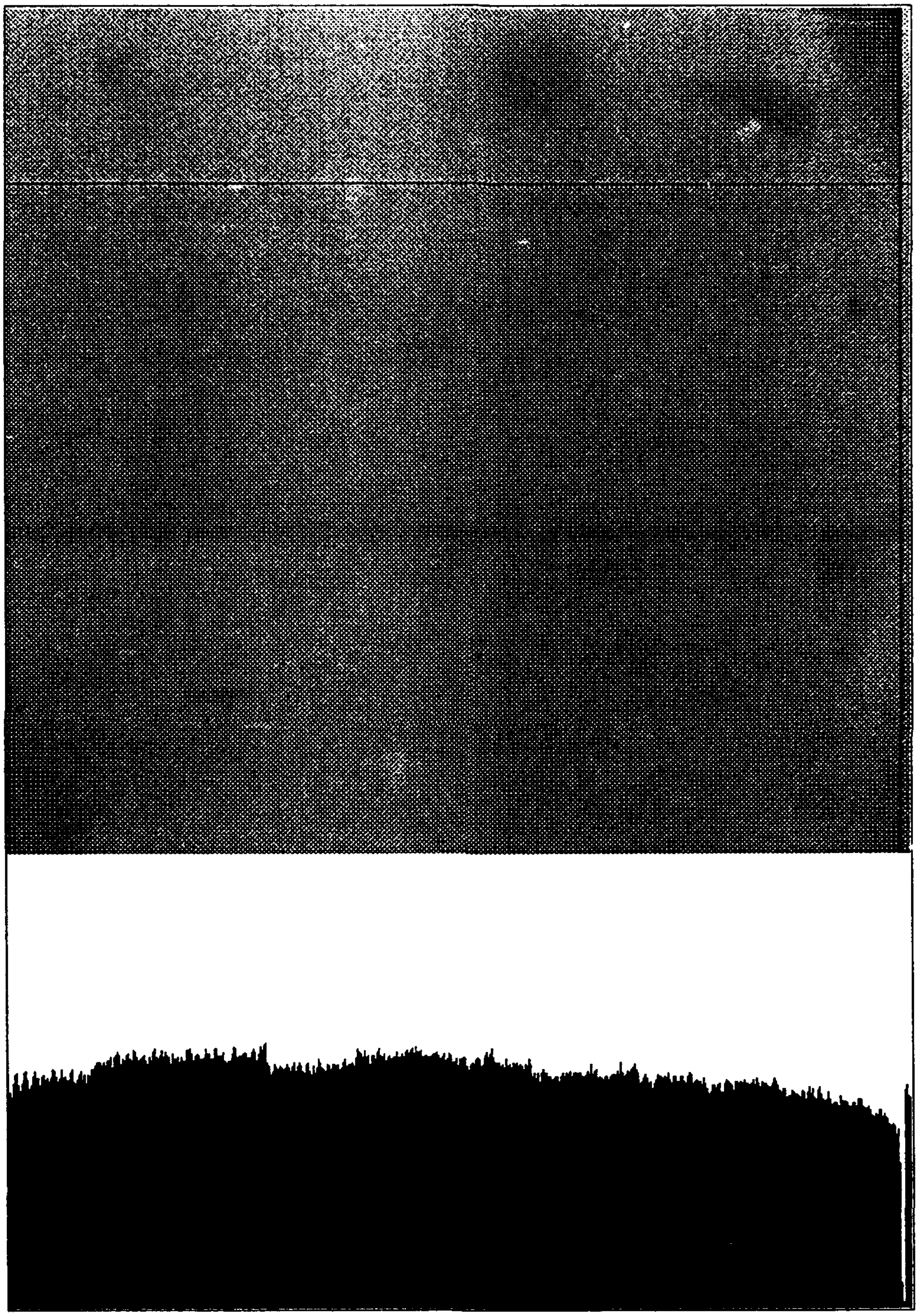


Figure 18 Case 15. Rectangles in Noise

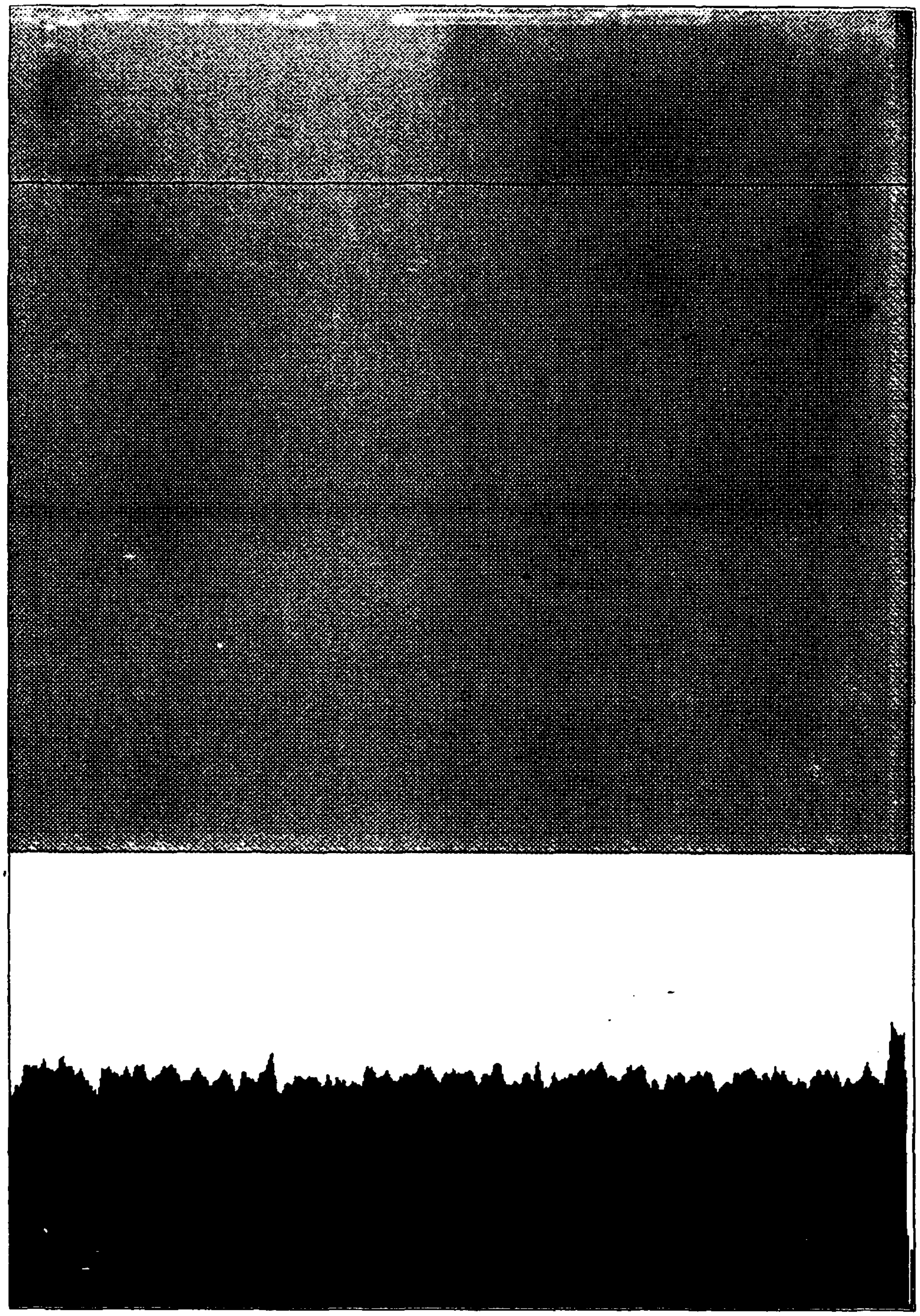


Figure 19 Case 16. Processed Rectangles In Noise

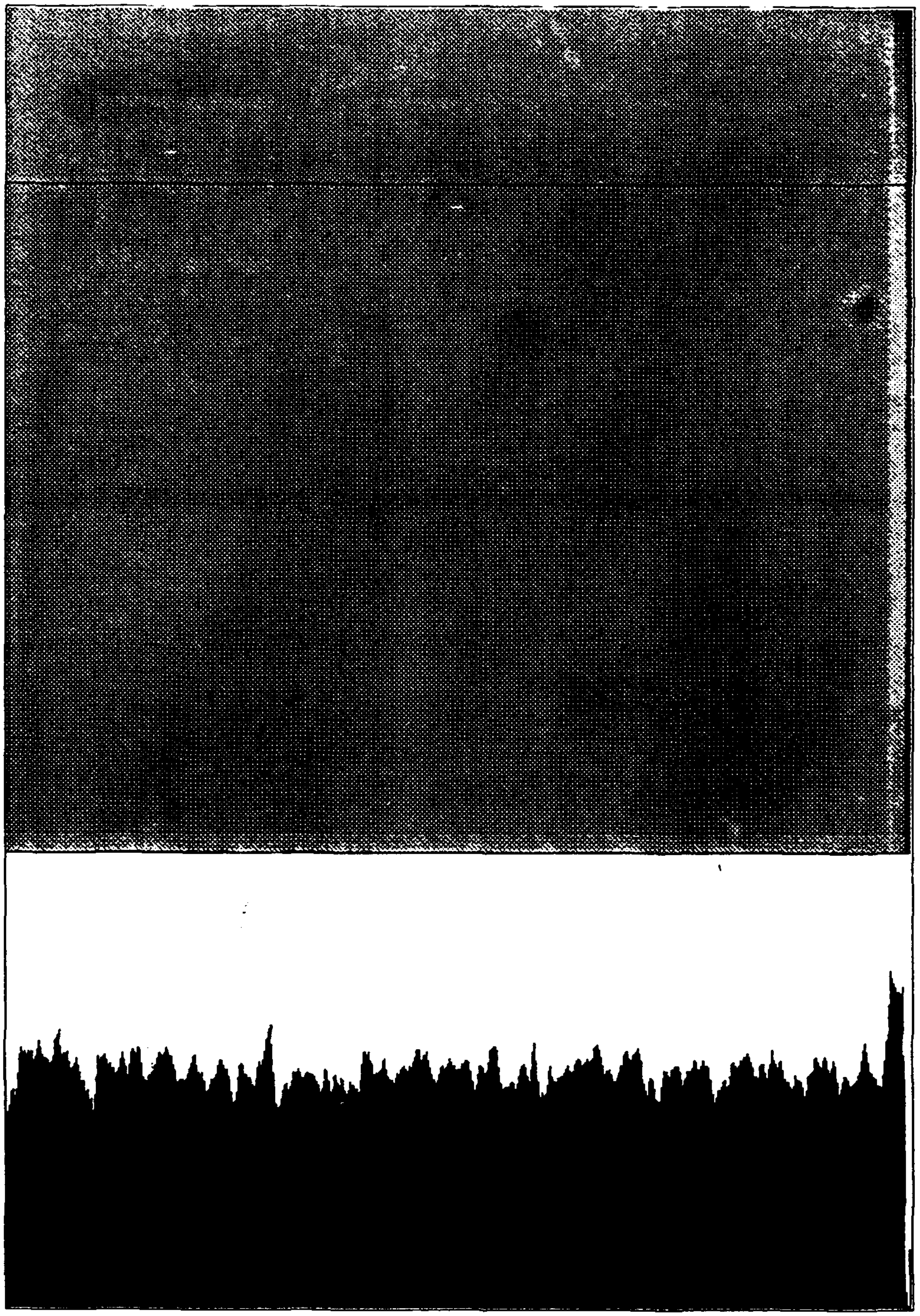


Figure 20 Case 17. Processed Rectangles in Noise, Increased Exponent

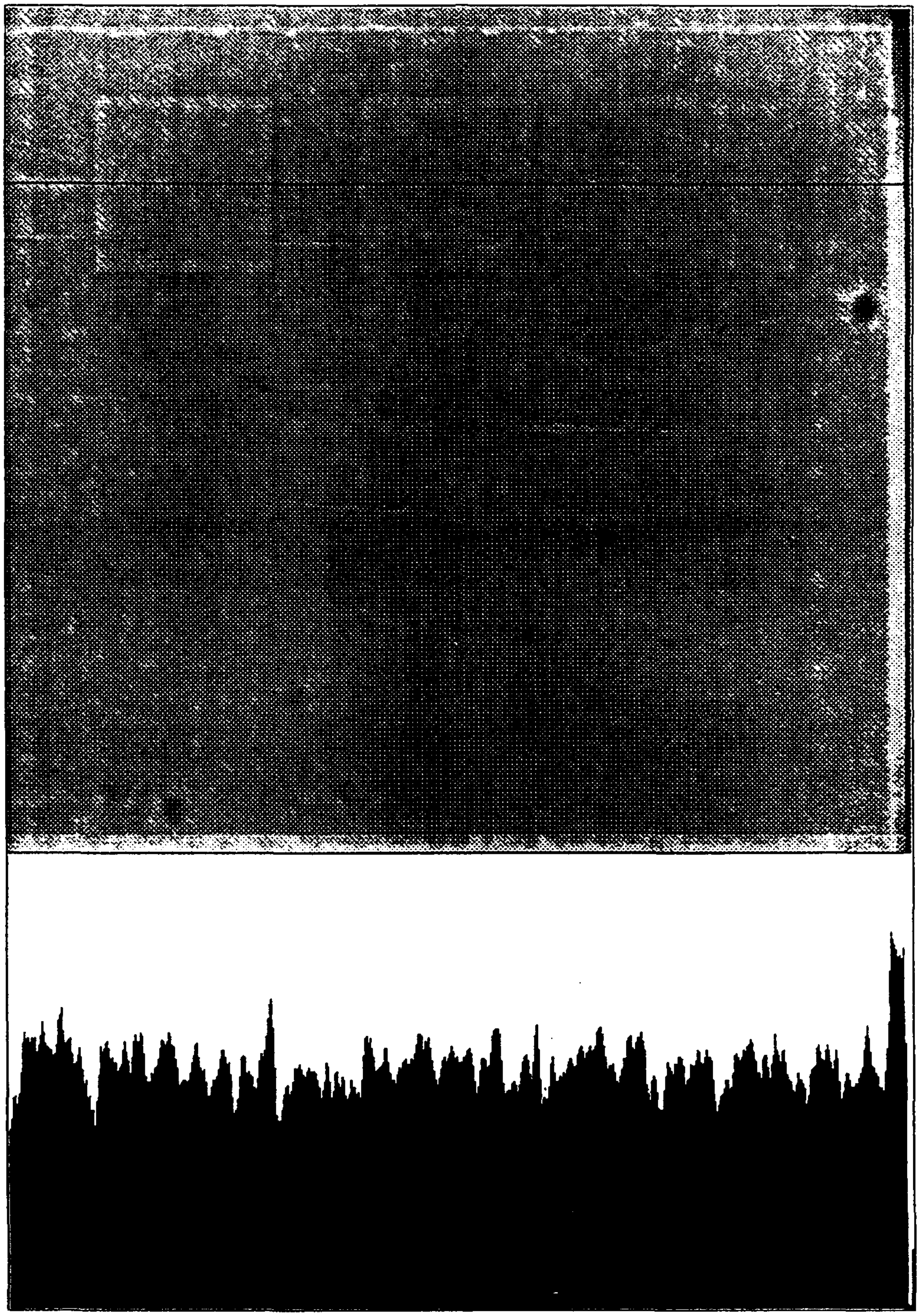


Figure 21 Case 18. Processed Rectangles in Noise, Increased Exponent

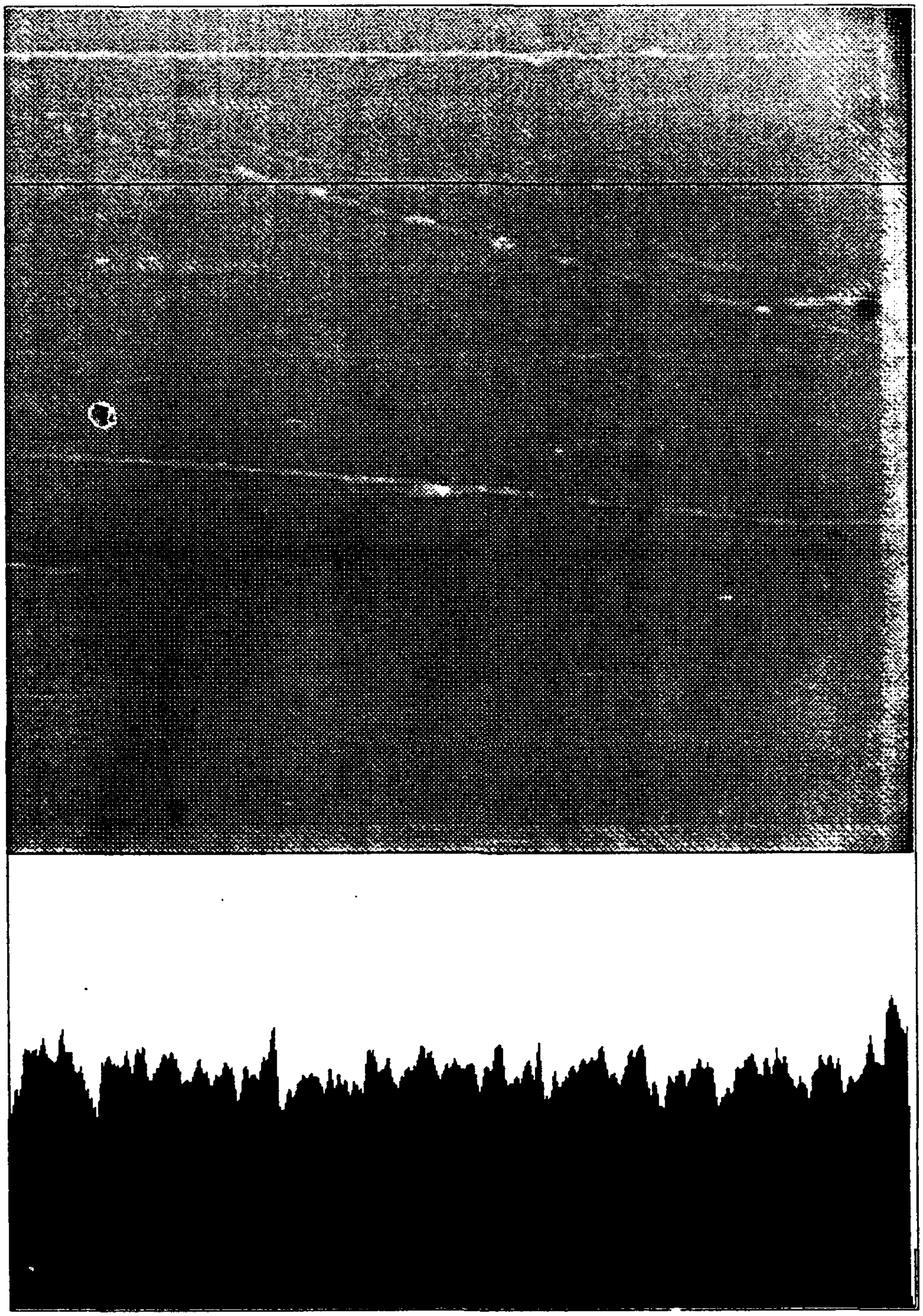


Figure 22 Case 19. Processed Rectangles In Noise, Large Inhibition Region

excitation and inhibition regions constant. These results are consistent with the previous cases.

Figure 22 shows the effect of increasing the size of the inhibition region. The noise level is decreased, since the larger inhibition region averages the noise over a larger area. Because of this result, another test case was run with an even larger inhibition region. The result, shown in Figure 23, shows the noise reduced slightly more.

Figure 24 shows what happened when the image was processed with a small excitation region and a small inhibition region. The borders became very thin and hard to see. With a larger excitation region, the noise is reduced because of averaging over a larger area, but the image is harder to see, as shown in Figure 25. With a large excitation region and a large inhibition region, the rectangles become easier to see, as shown in Figure 26.

Figures 27 and 28 show the same images processed with a small excitation region, increasingly large inhibition regions, and large exponents. The noise is still reduced, but the clarity of the image is improved. Subjectively, this type of filtering does the best job of extracting the squares from noise.

C. Shape Filtering

The previous section showed how the retina model can be manipulated to change the output. Based on this information, a filter which can amplify objects of a certain size and

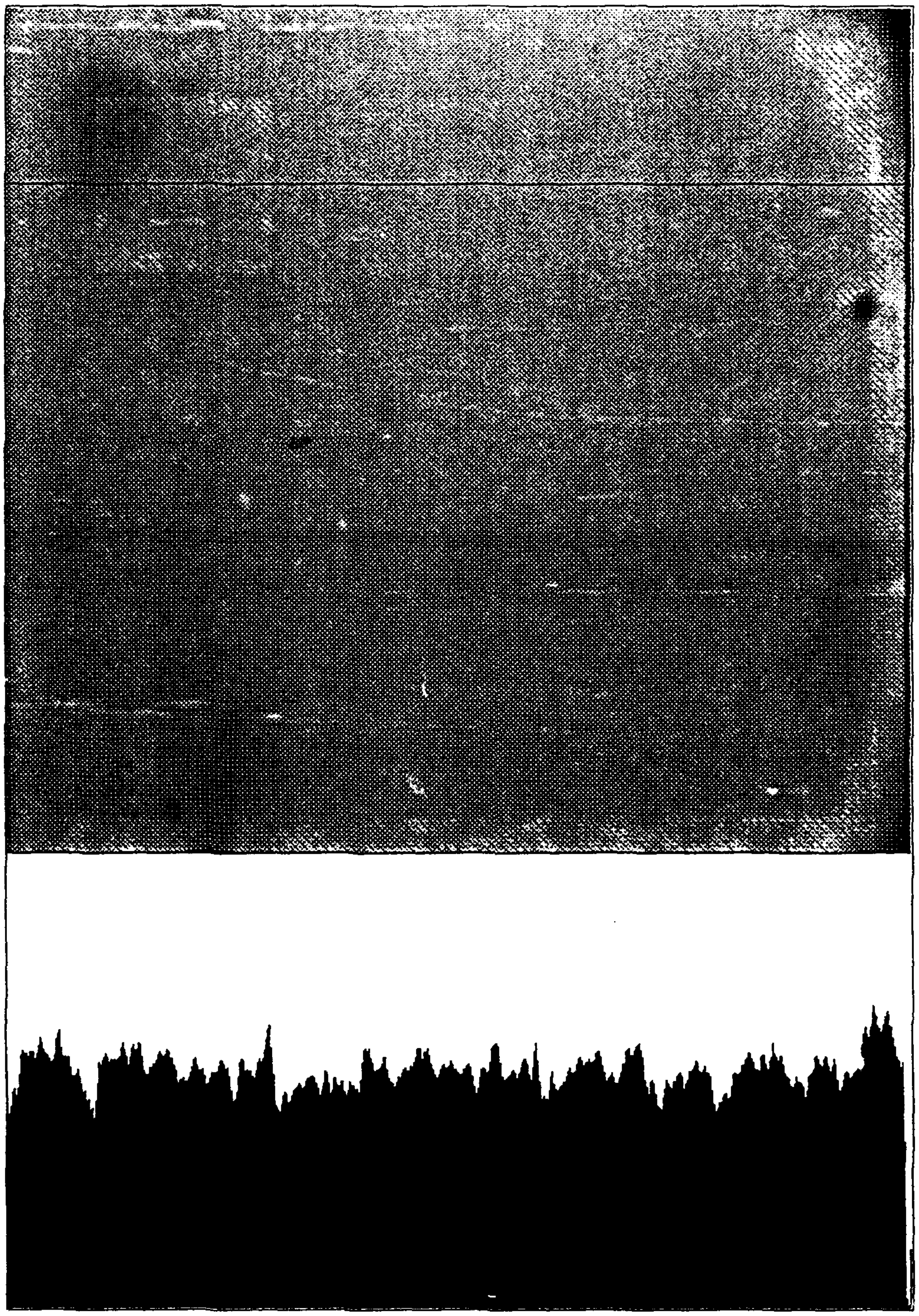


Figure 23 Case 20. Processed Rectangles In Noise, Larger Inhibition Region

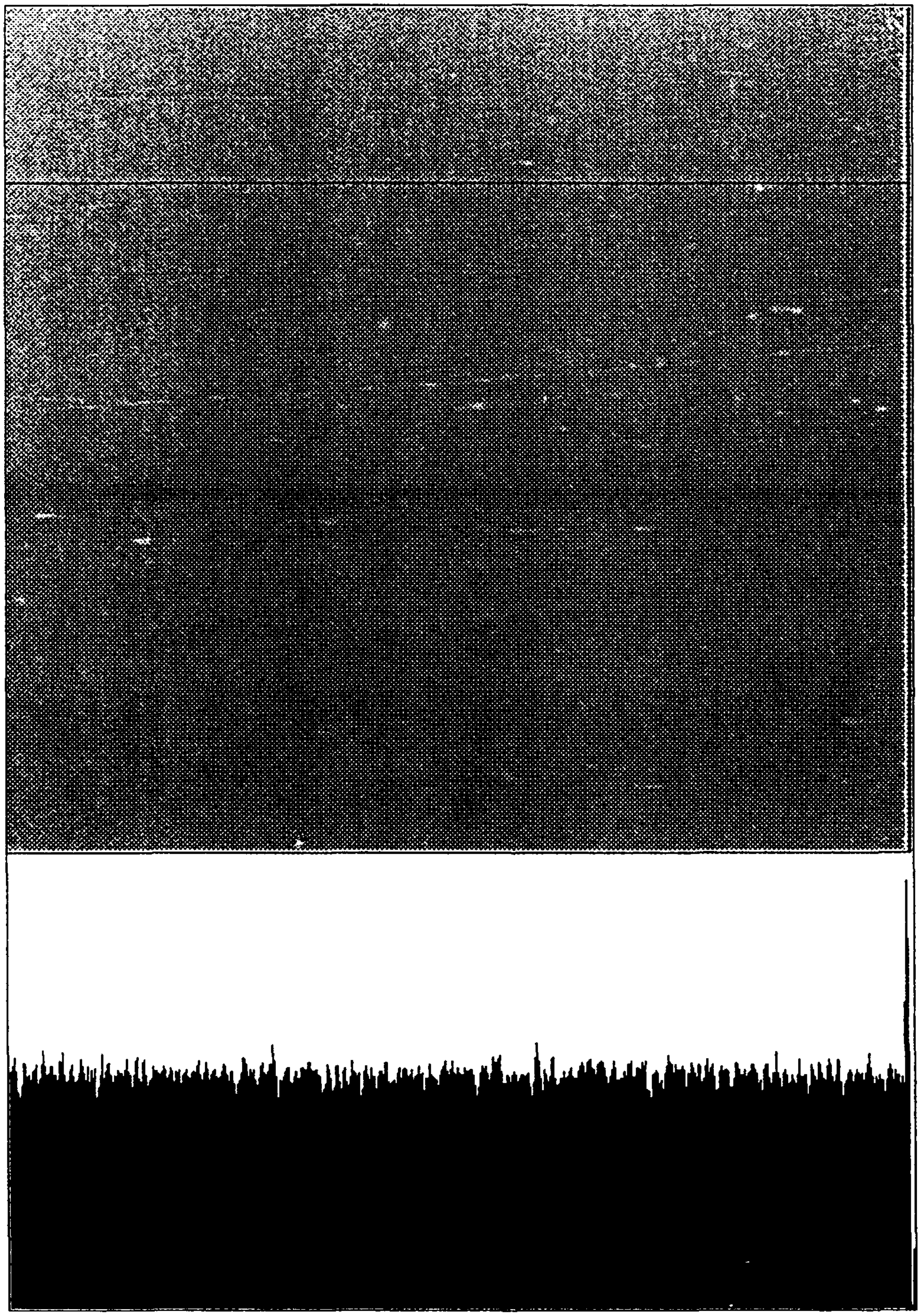


Figure 24 Case 21. Processed Rectangles In Noise, Small Excitation and Inhibition Regions

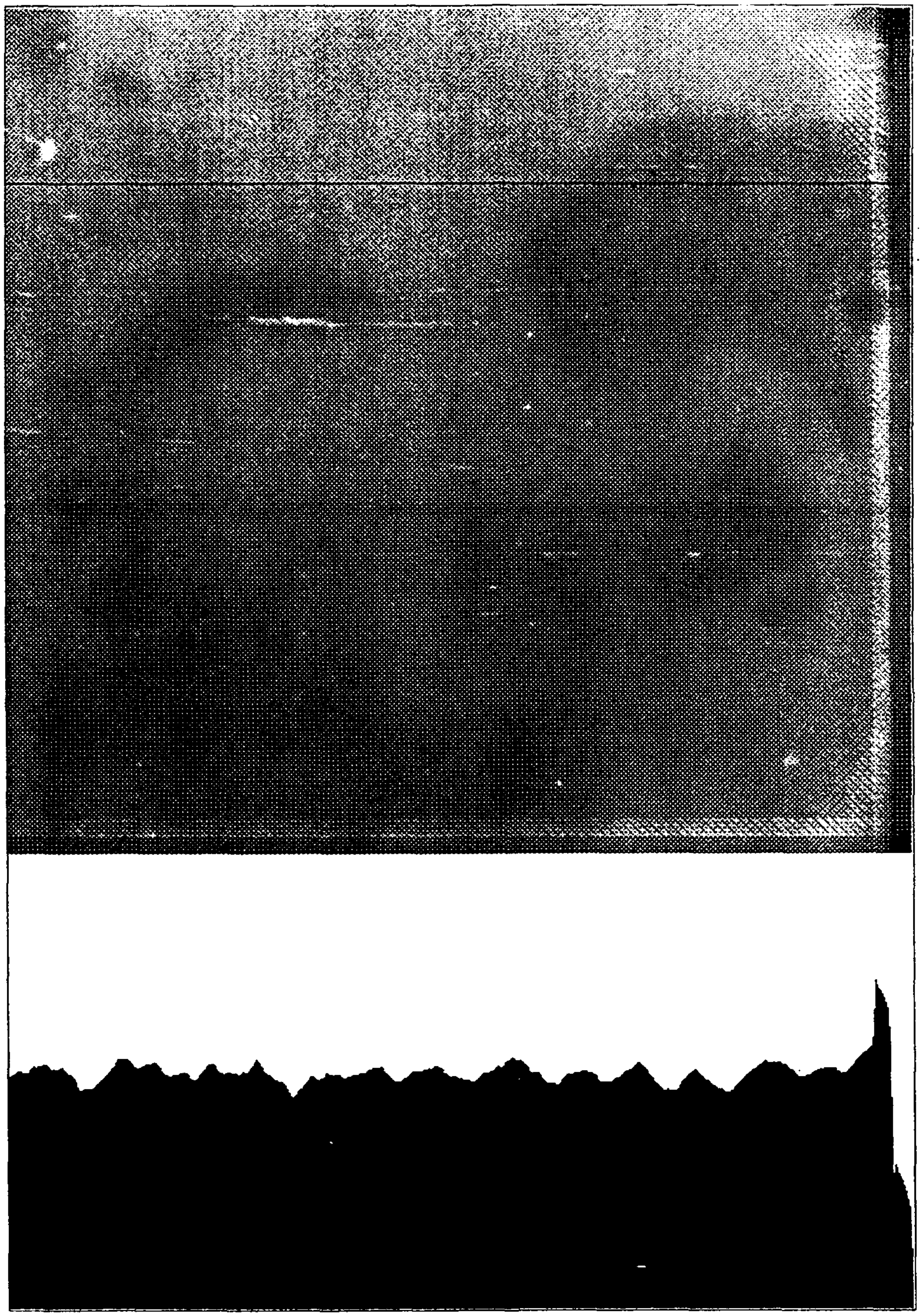


Figure 25 Case 22. Processed Rectangles In Noise, Large Excitation Region, Small Inhibition Region

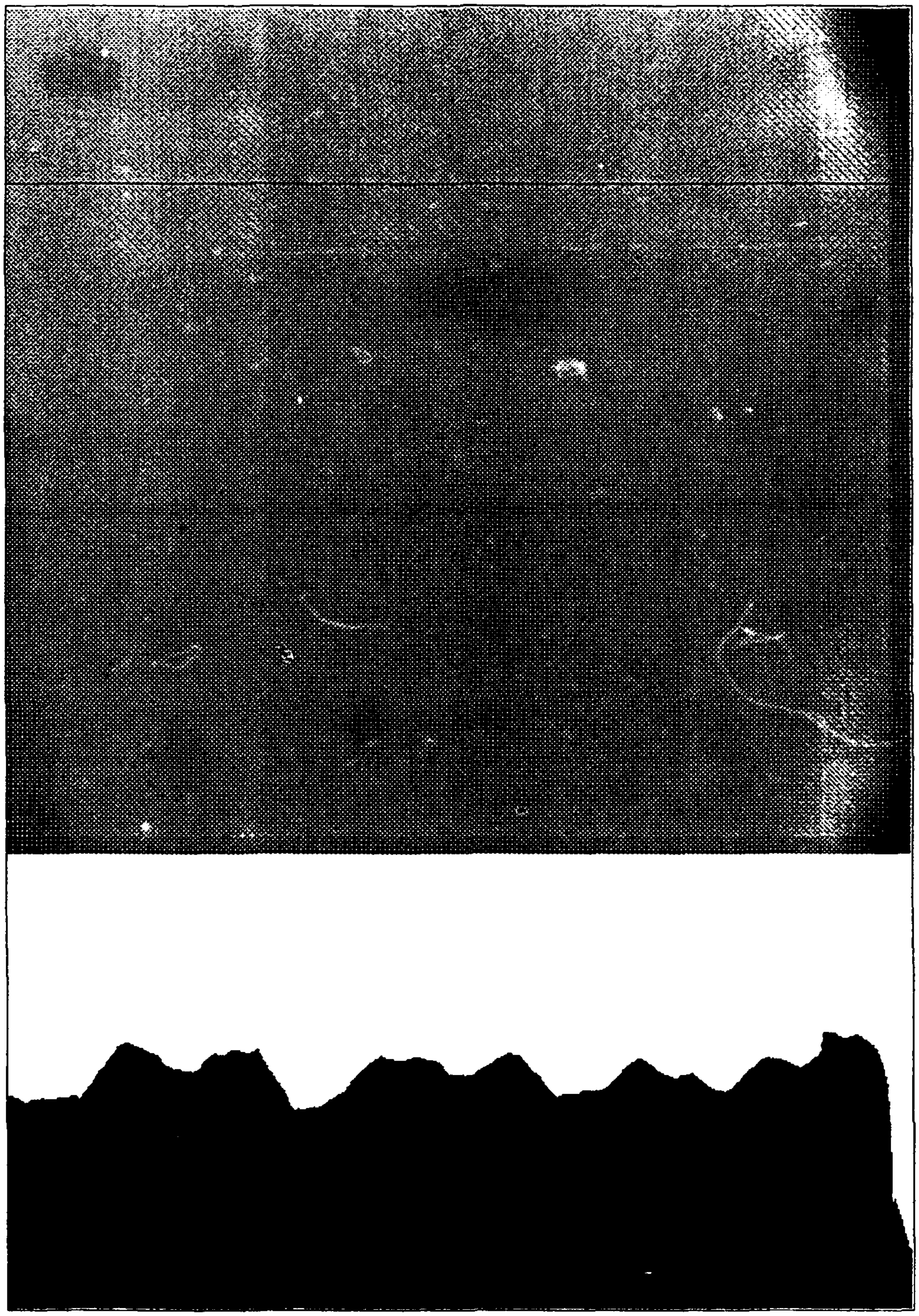


Figure 26 Case 23. Processed Rectangles In Noise, Large Excitation and Inhibition Regions

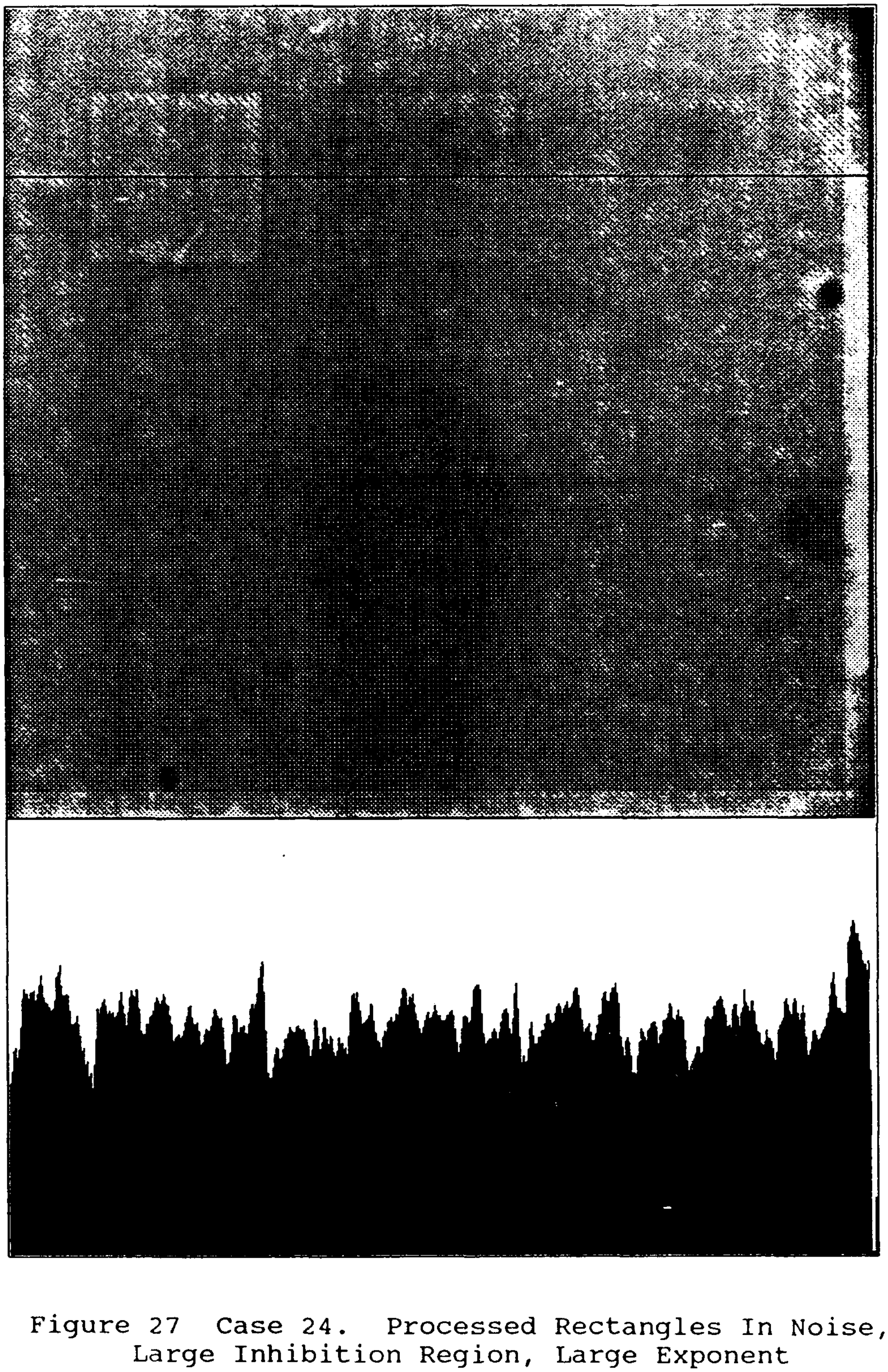


Figure 27 Case 24. Processed Rectangles In Noise, Large Inhibition Region, Large Exponent

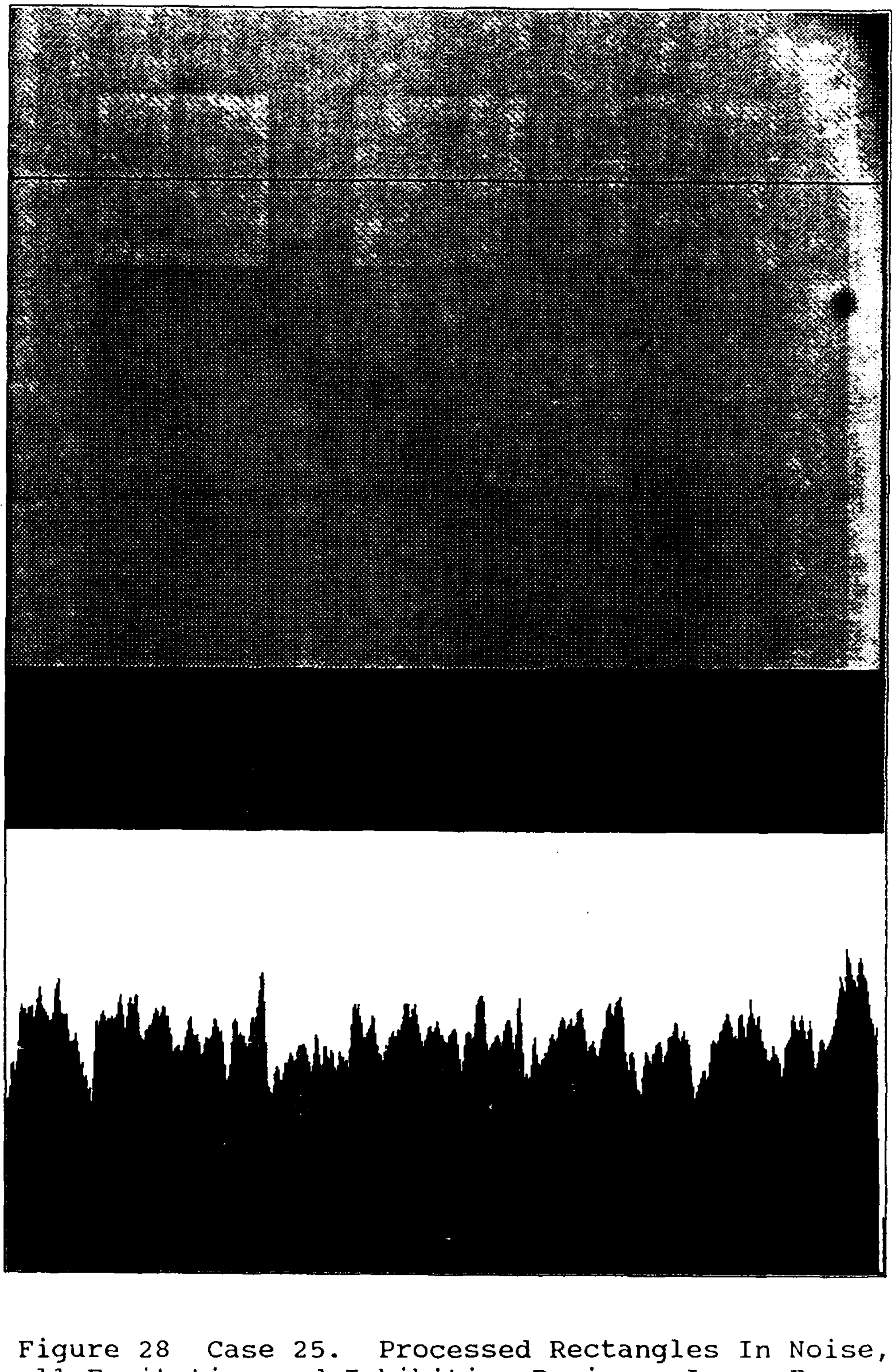


Figure 28 Case 25. Processed Rectangles In Noise, Small Excitation and Inhibition Regions, Large Exponent

shape, while attenuating all other objects, might be developed. The size and shape of objects which can pass through the filter could be adjusted by a careful choice of input parameters.

If the excitation and inhibition regions are much larger than the object of interest, the object will be attenuated because its signal will be averaged in with the background. If the excitation and inhibition regions are made very small, edge enhancement will occur on all medium and large objects in the field-of-view.

A reasonable hypothesis, therefore, is that if the size of the excitation region was approximately the size of the object of interest, that object would pass through the filter. Small objects would blend into the noise, and large objects would appear as a pattern of lines.

To investigate the ability of the retina model to filter objects of different shapes, a scene made up of three squares of different sizes was constructed (Figure 29). The large square is 145 x 145 pixels, the medium square is 41 x 41 pixels, and the small rectangle is 9 x 9 pixels. The squares are all at the same intensity.

Figure 30 shows the results of applying a retina filter with excitation region the same size as the medium rectangle, and the inhibition region slightly larger. Figure 31 shows the result of a filter with the same excitation region, but a larger inhibition region.

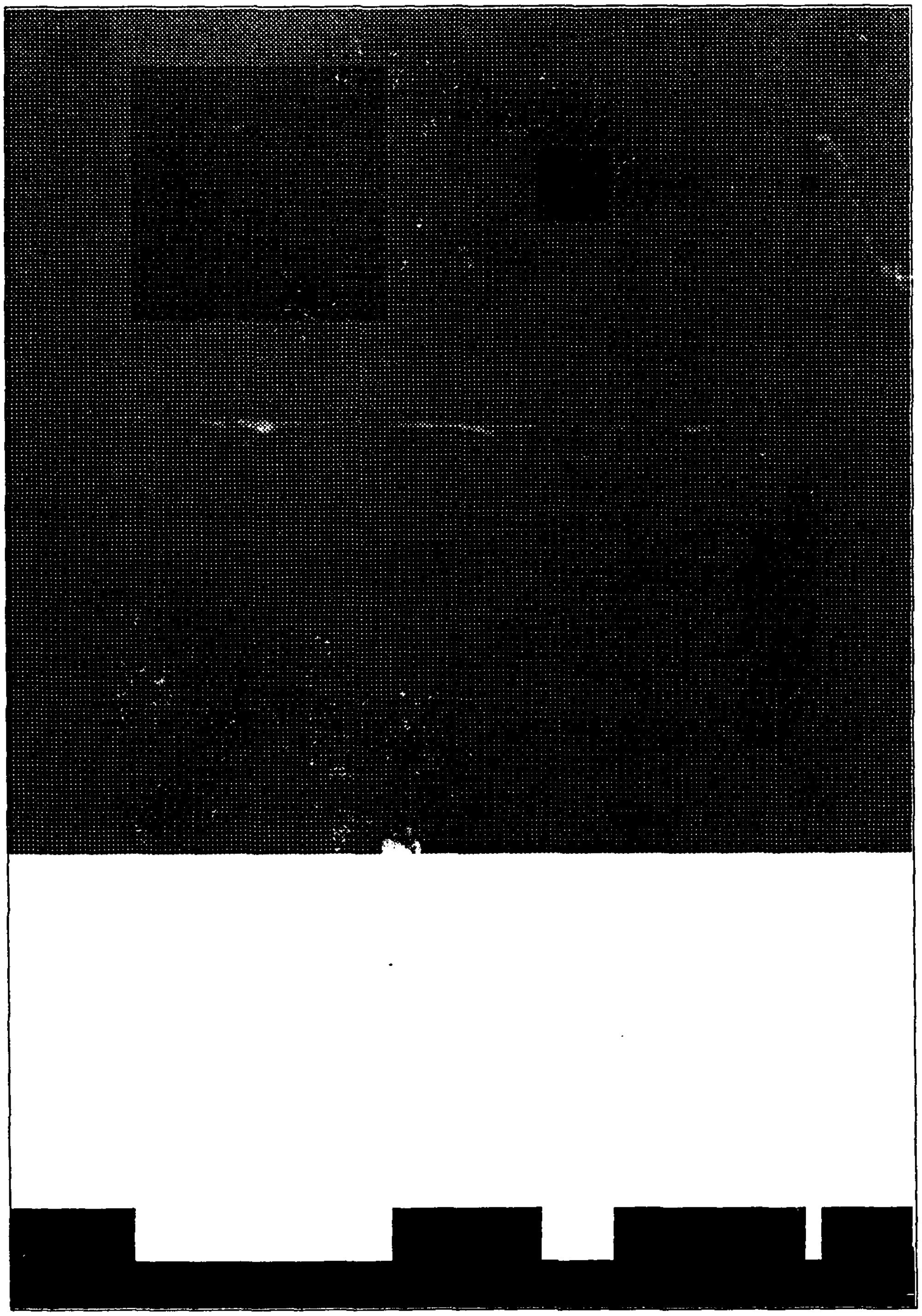


Figure 29 Case 26. Three Squares

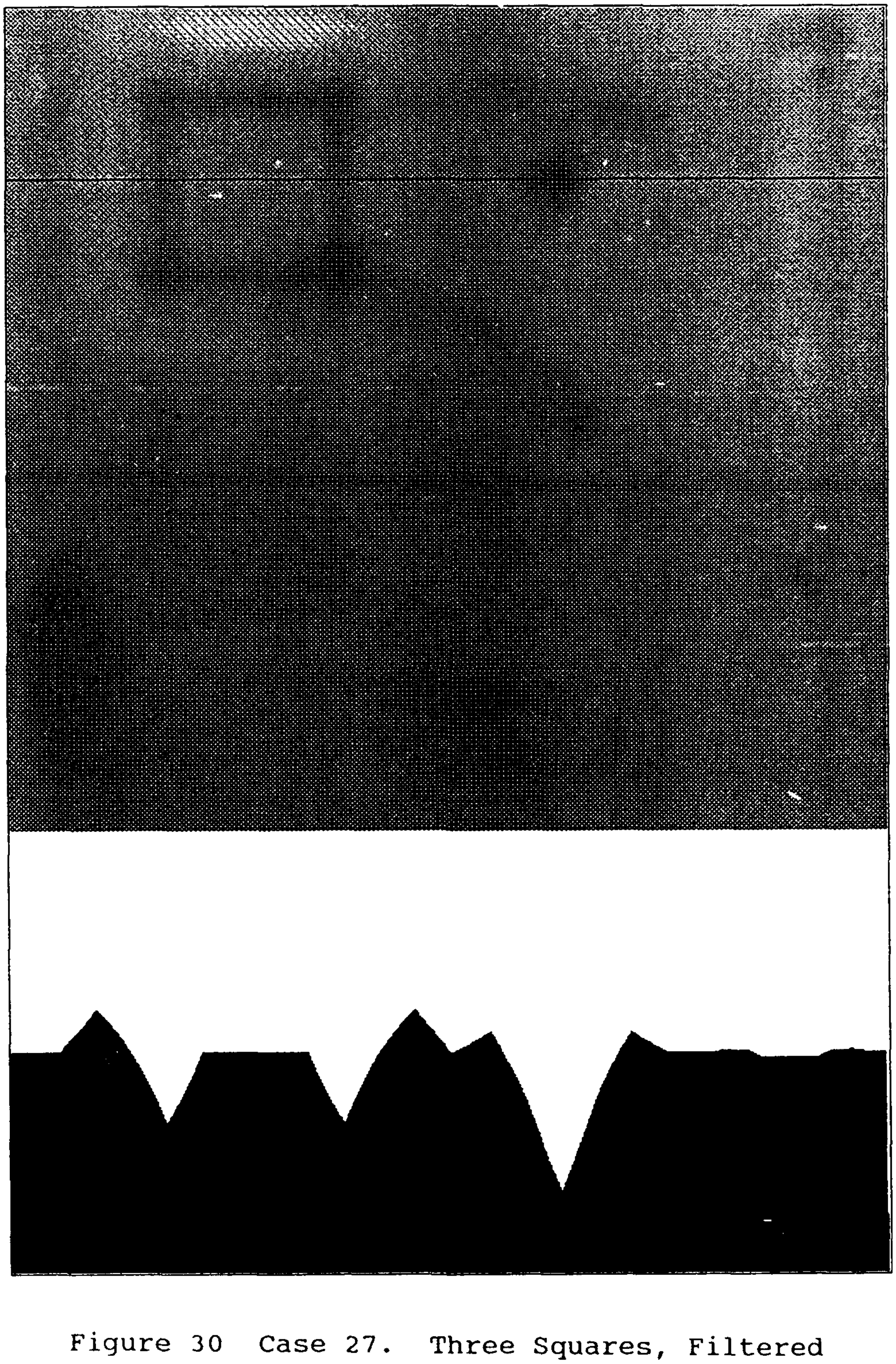


Figure 30 Case 27. Three Squares, Filtered

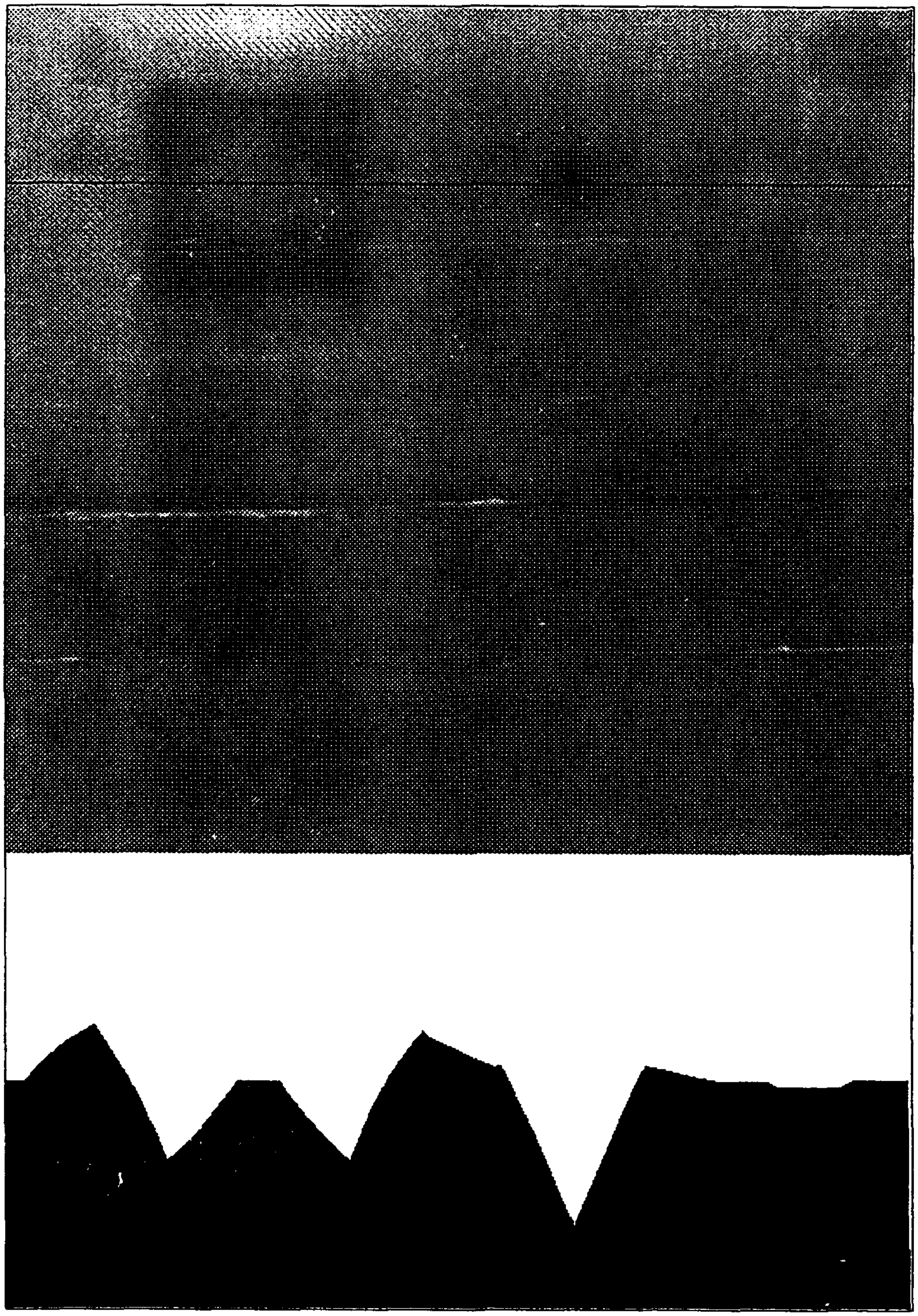


Figure 31 Case 28. Three Squares, Filtered, Larger Inhibition Region

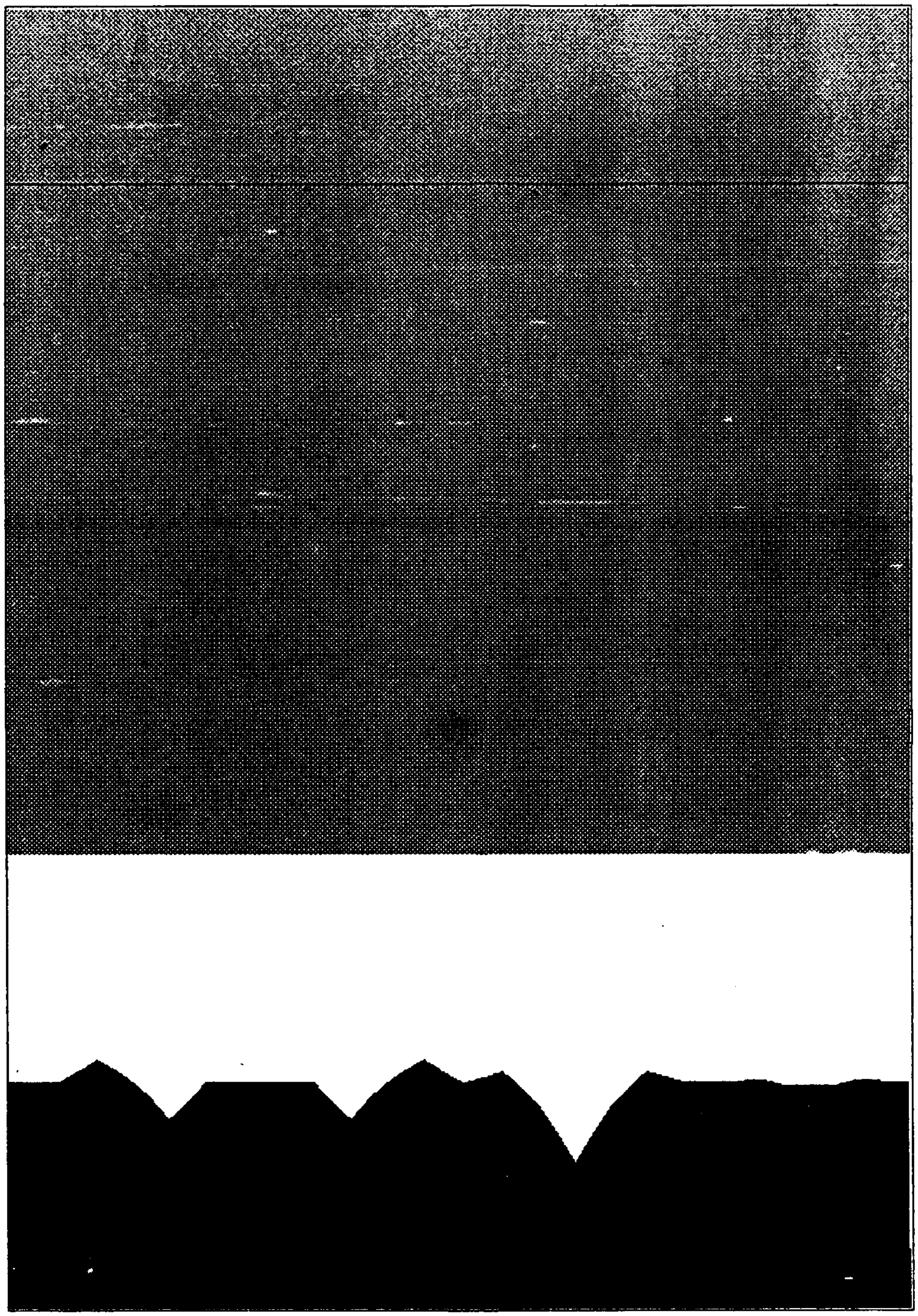


Figure 32 Case 29. Three Squares, Filtered, Small Exponent

Figure 32 has the same parameters as Figure 30, but with a smaller exponent.

In each case, the medium rectangle has a greater degree of enhancement than the others. Increasing the size of the inhibition region spreads out the peaks and valleys, and also increases their magnitudes. Decreasing the exponent decreases the magnitude of the peaks and valleys.

A set of tests was then run to determine if rectangles of the same size and different orientations could be distinguished. The input image is shown in Figure 33. Each rectangle is 31 x 11 pixels, but they are placed orthogonally in the image.

The output of a retinal filter with an 11 x 31 pixel excitation region is shown in Figure 34. The 11 x 31 rectangle is much darker than the 31 x 11 rectangle. The distinction is further enhanced when the exponent is raised from '1' to '2', as shown in Figure 35.

Figure 36 shows the same input as Figure 34, but with rectangles at a higher intensity level than the background. The input parameters for Figures 37 and 38 are the same as for Figures 35 and 36, and the results are the same, except the output images are brighter instead of darker.

Figure 39 shows a digitized image of a tank. The retinal model was set with an excitation region approximately the same size as the targetter at the base of the barrel and the wheels. Figure 40 shows the output of

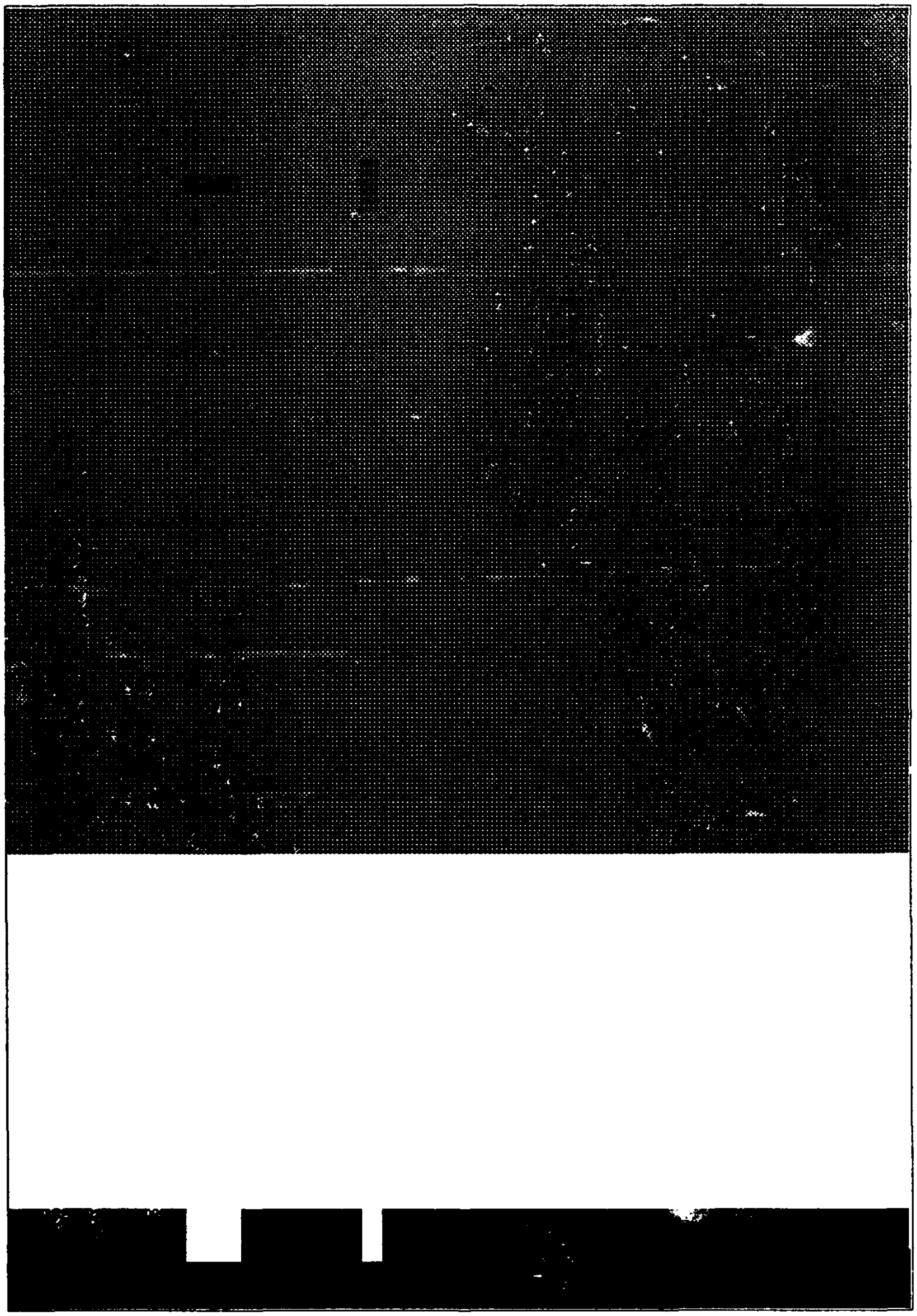


Figure 33 Case 30. Orthogonal Rectangles

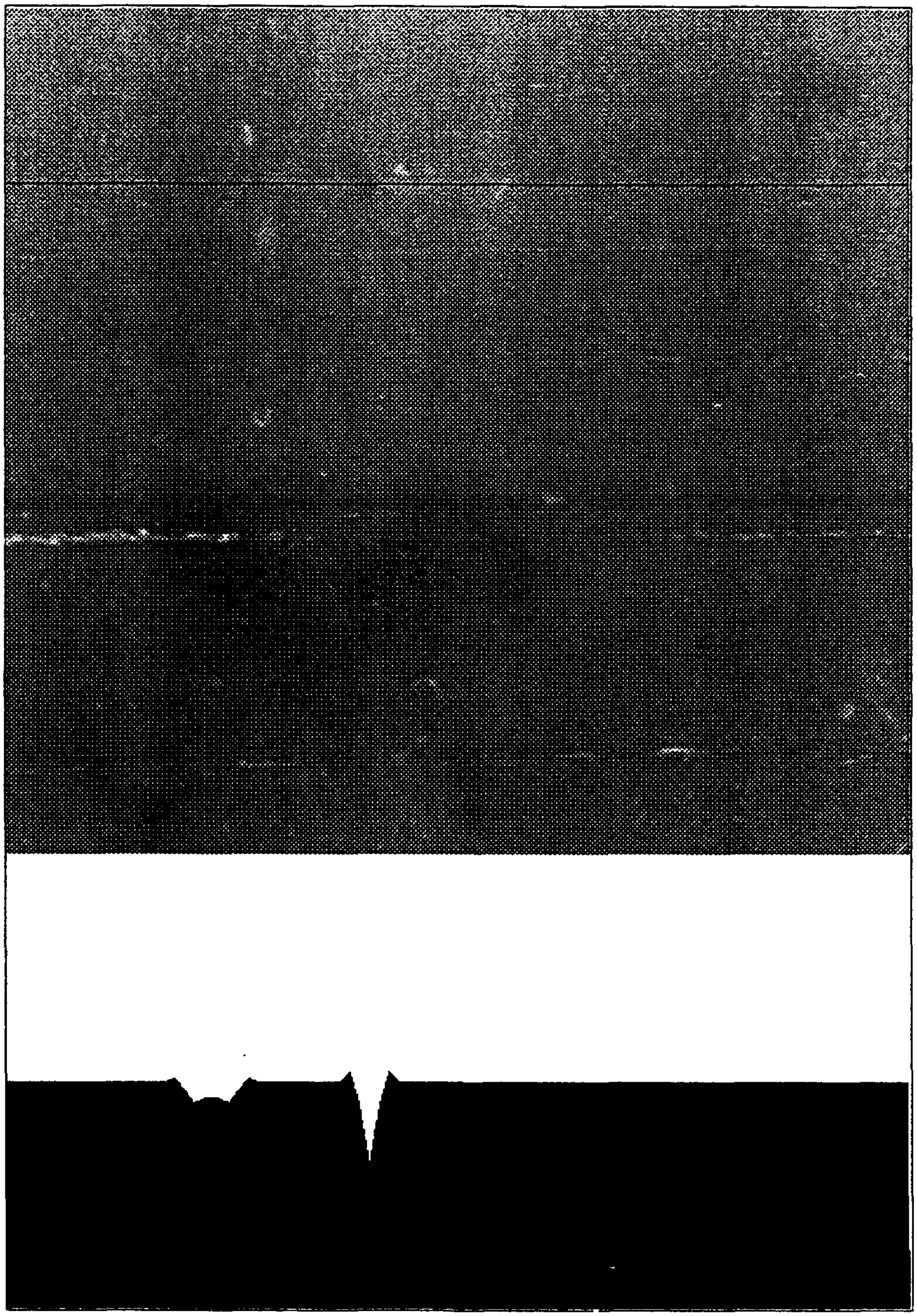


Figure 34 Case 31. Orthogonal Rectangles, Filtered

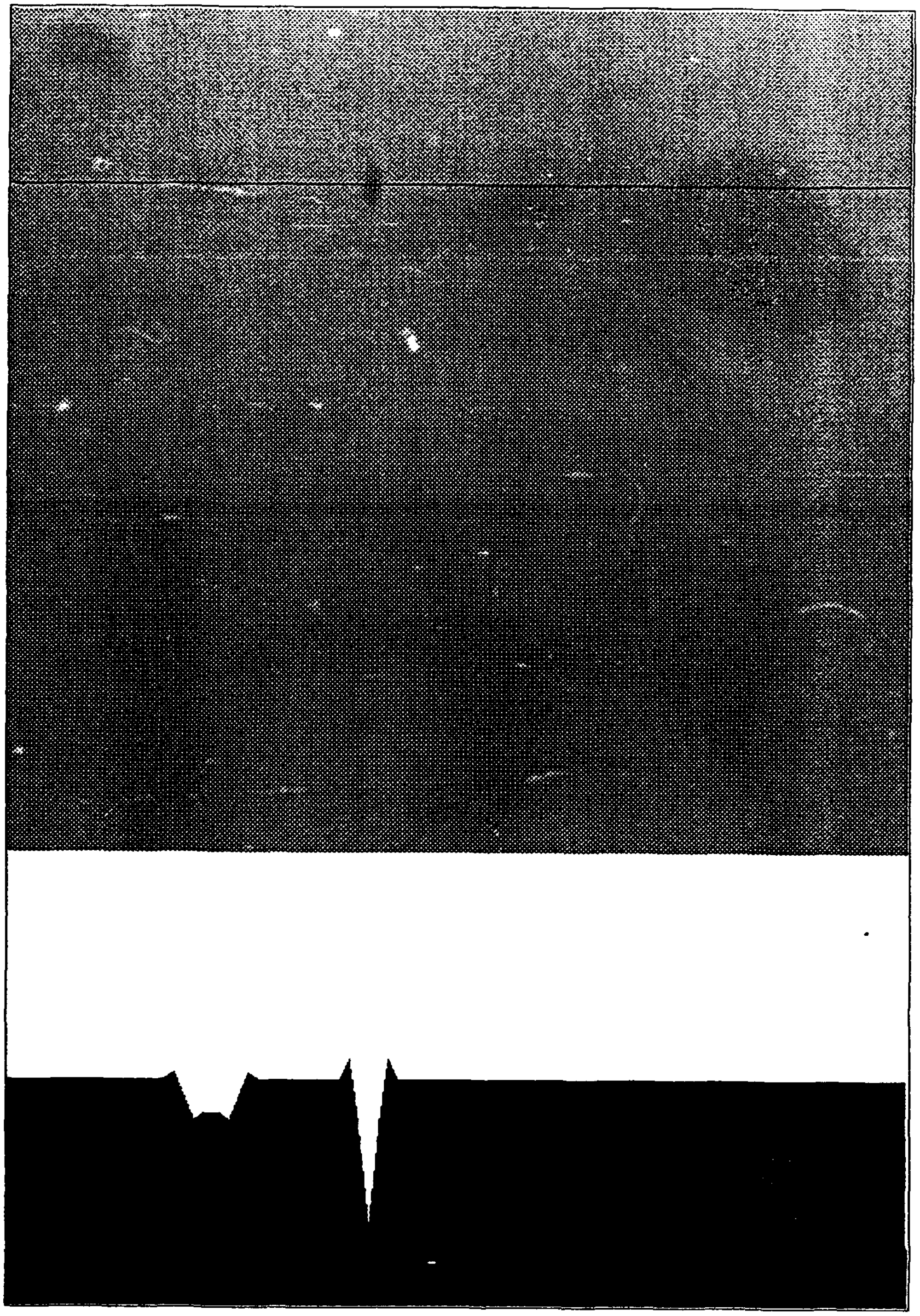


Figure 35 Case 32. Orthogonal Rectangles, Filtered, Larger Exponent

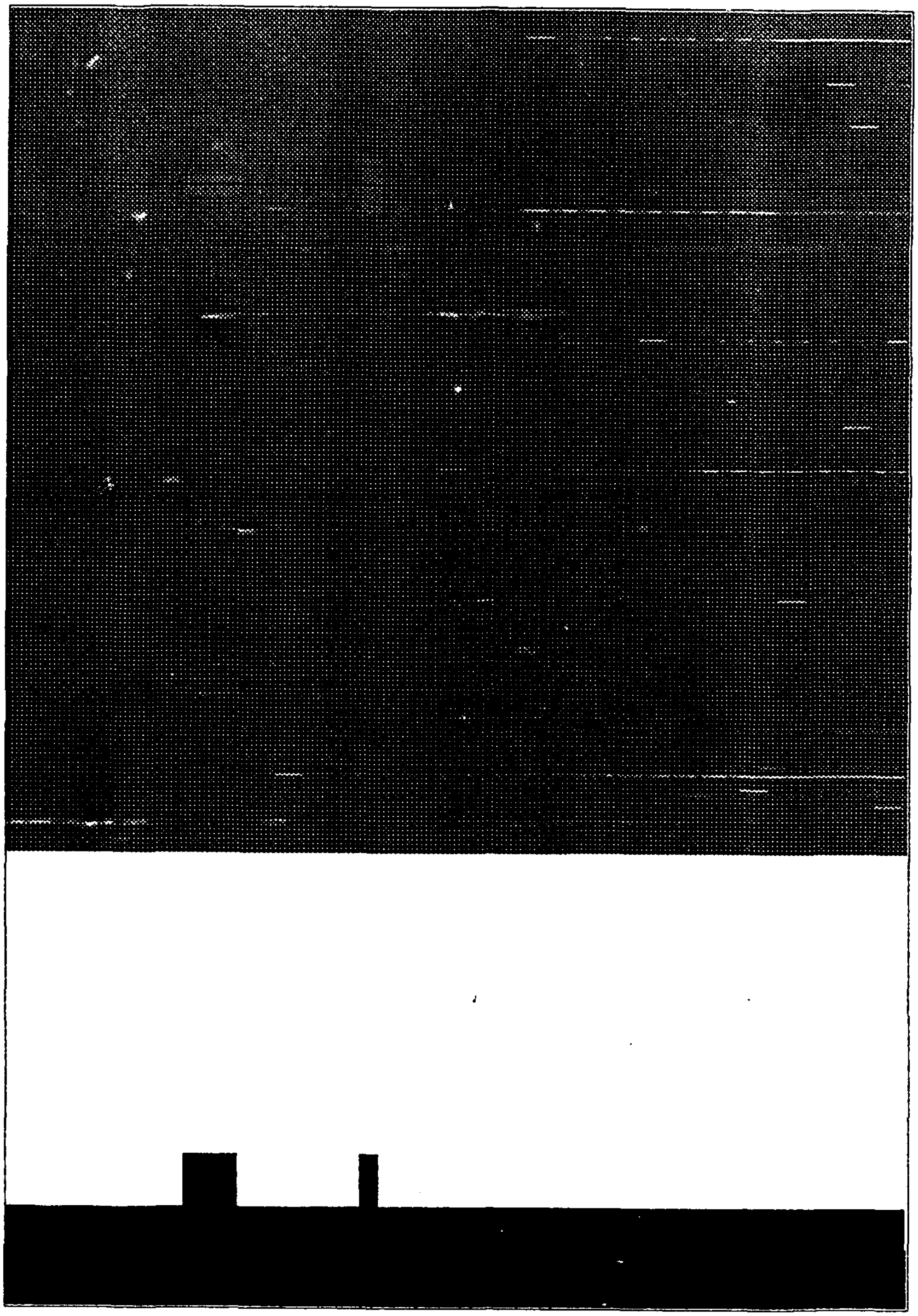


Figure 36 Case 33. Orthogonal Rectangles

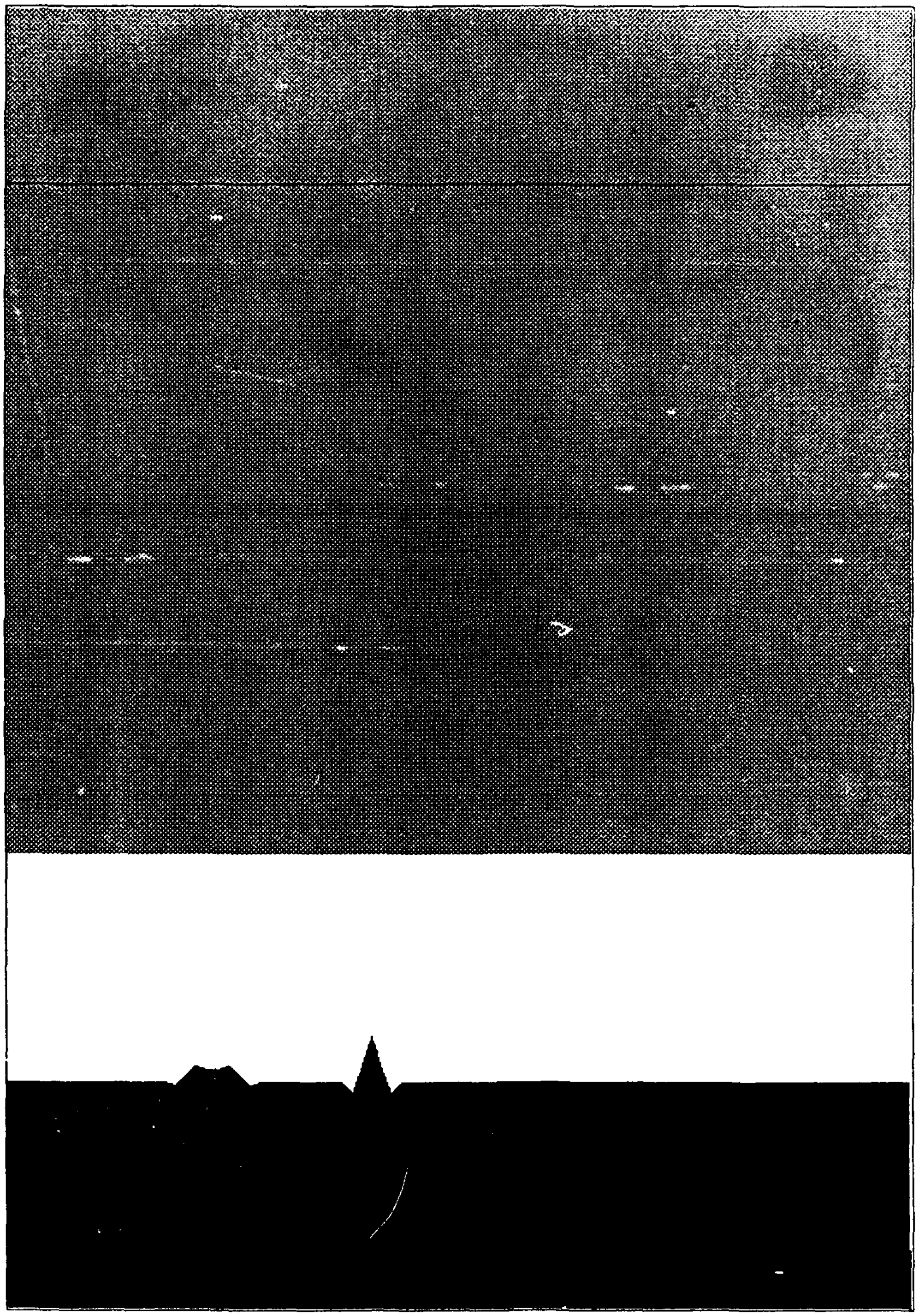


Figure 37 Case 34. Orthogonal Rectangles, Filtered

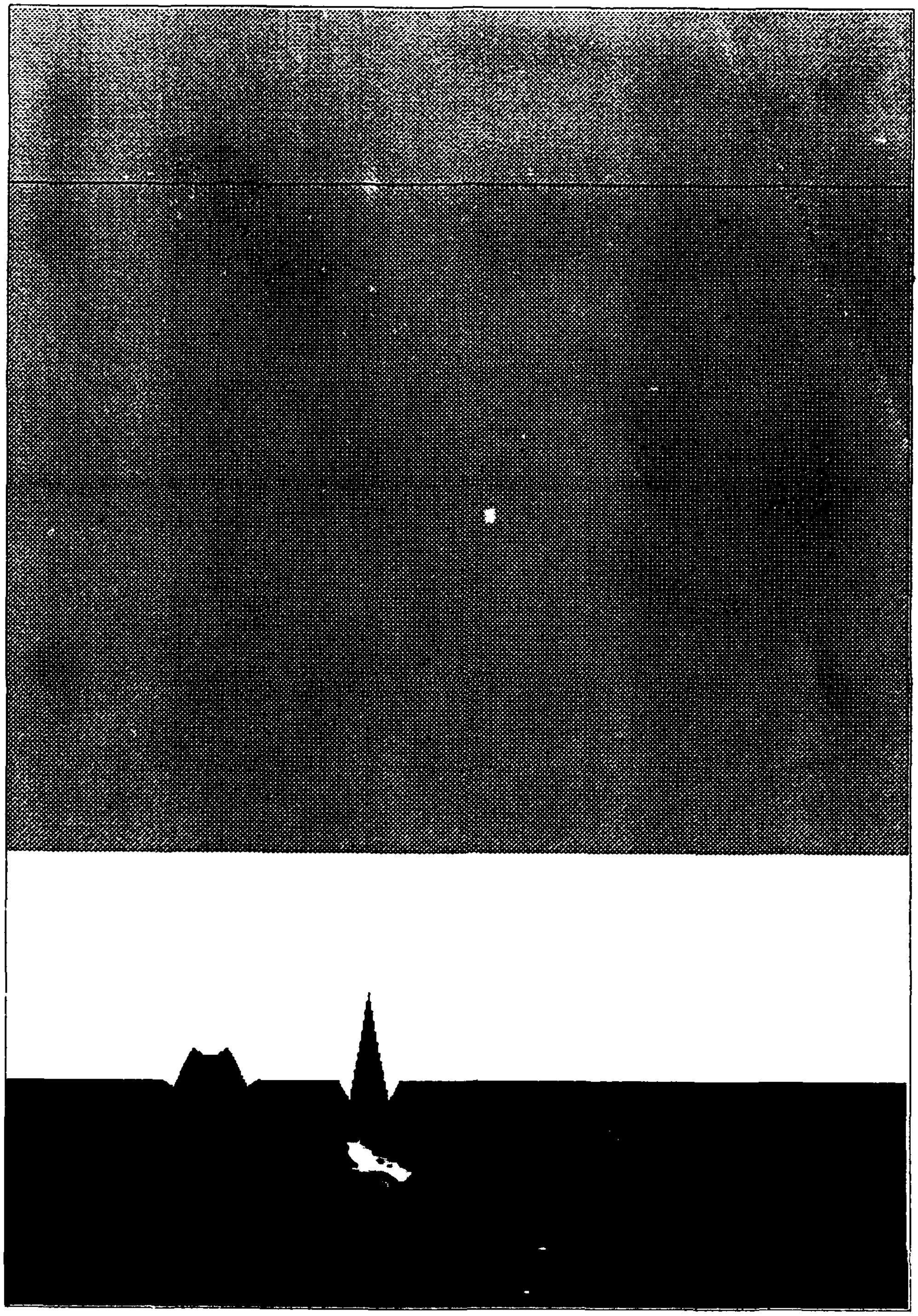


Figure 38 Case 35. Orthogonal Rectangles, Filtered, Larger Exponent

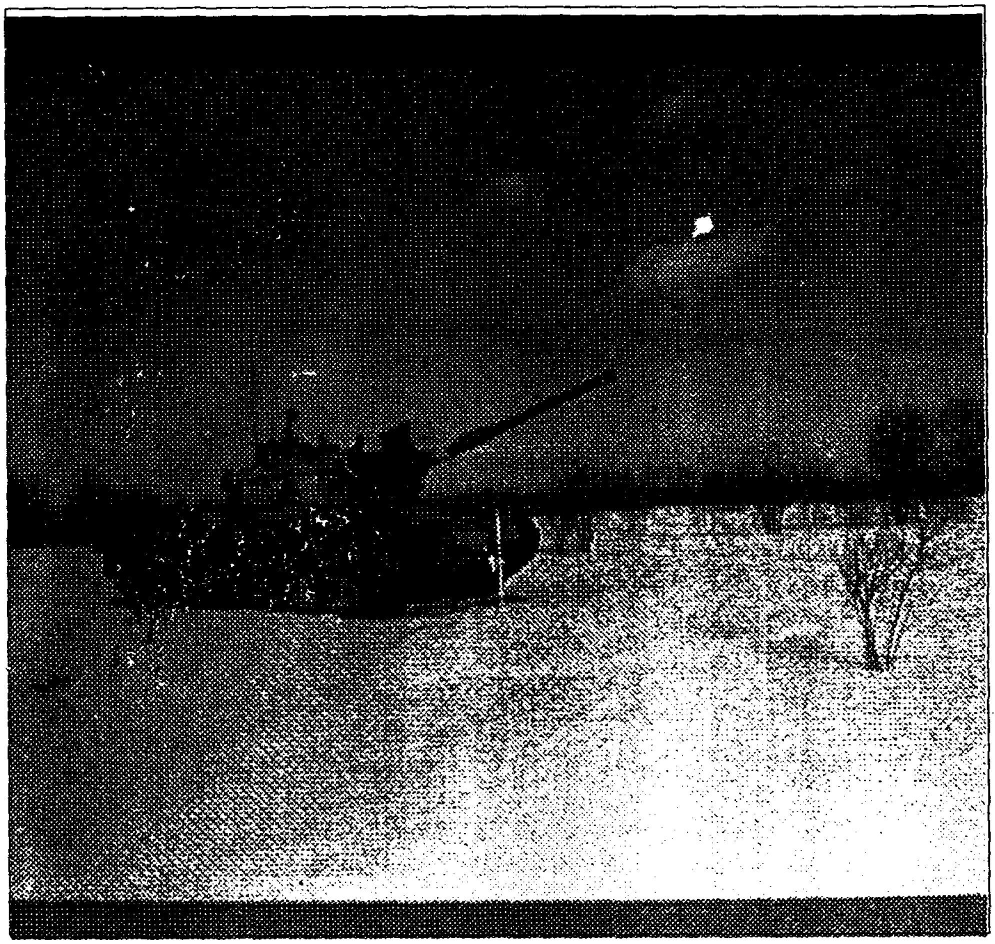


Figure 39 Case 36. Tank

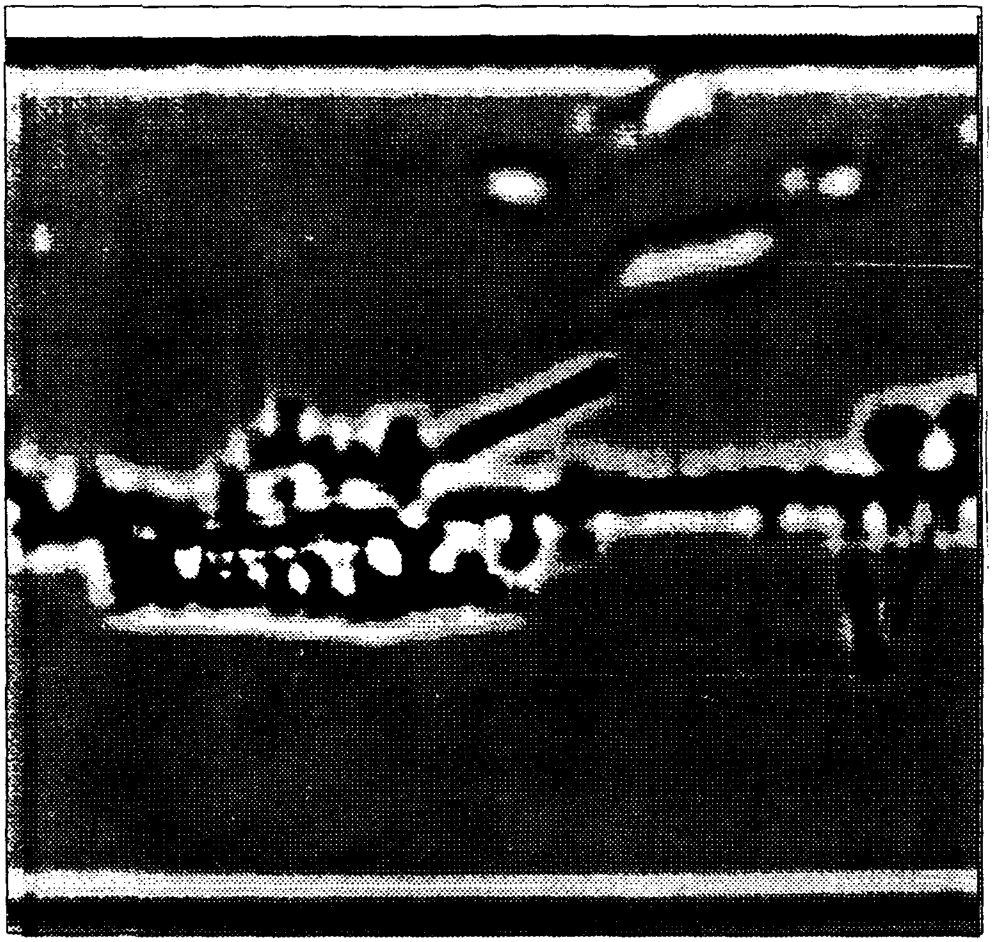


Figure 40 Case 37. Tank,
Targetter and Wheels Enhanced

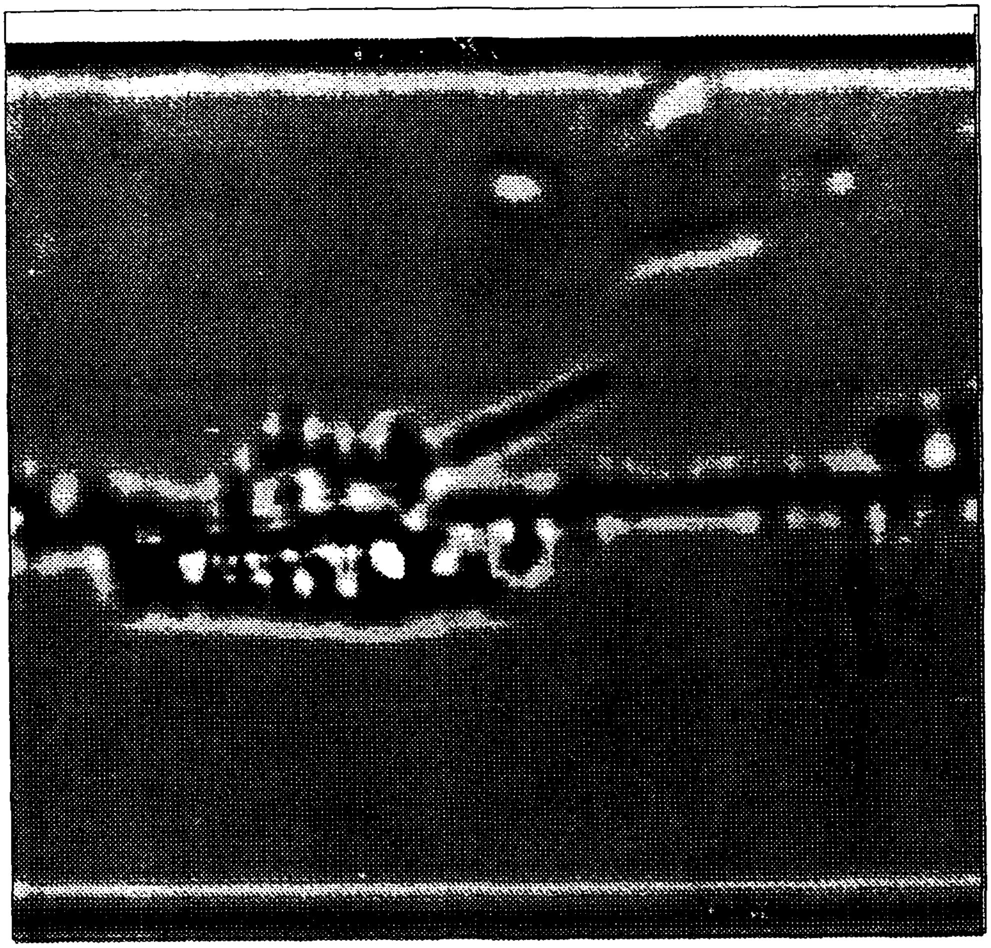


Figure 41 Case 38. Tank,
Targetter and Wheels Enhanced, Smaller Exponent

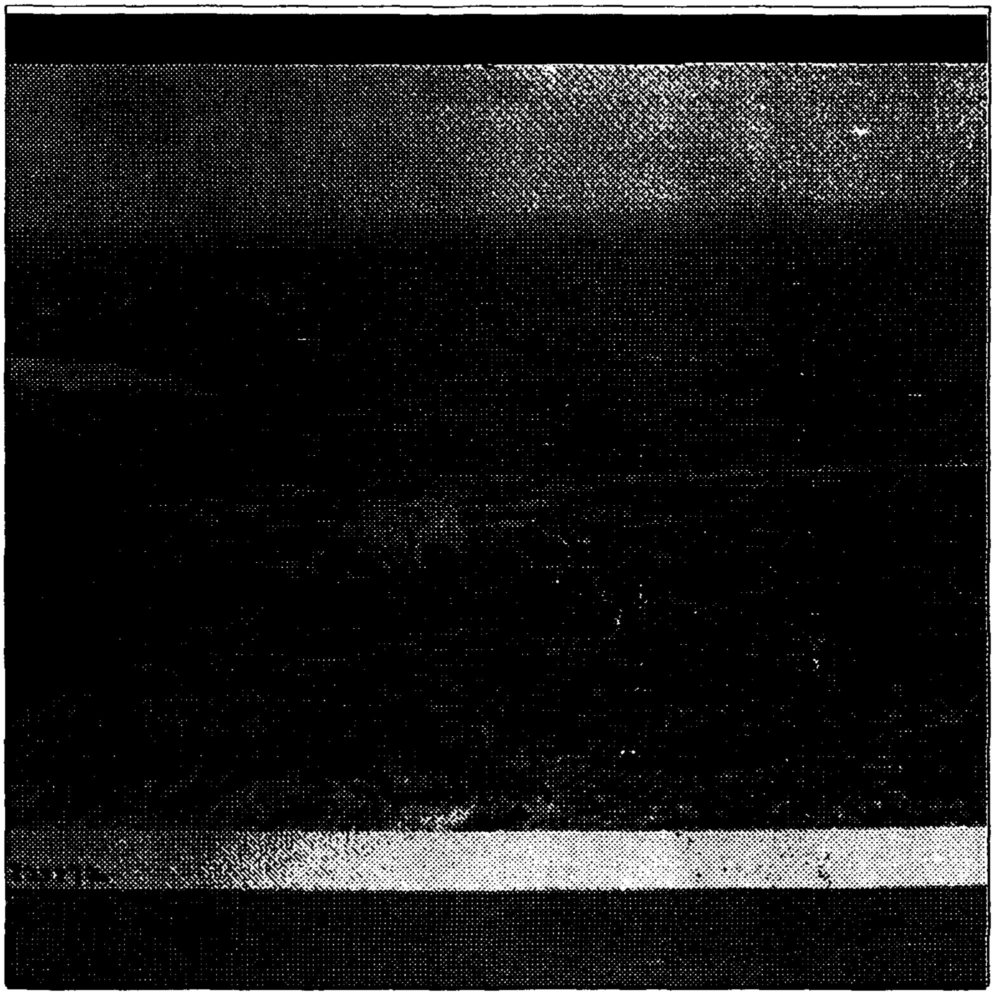


Figure 42 Case 39. Tank

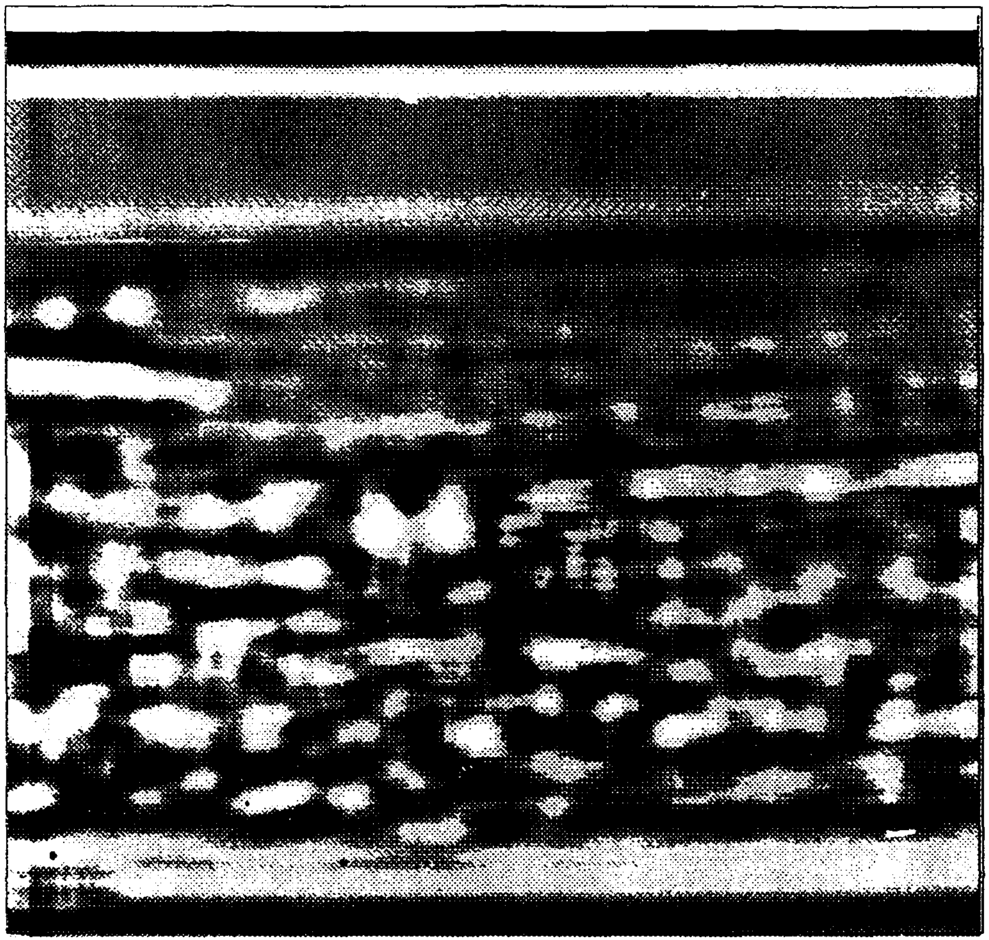


Figure 43 Case 40. Enhanced Tank .

the program using an exponent of 10, and Figure 41 show the result using an exponent of five. The targetter and the wheels were enhanced, while the other components of the tank, as well as the background, were minimized.

Figure 42 also shows a digitized image of a tank, but the tank is at a greater range, the contrast with the background is less, and the aspect angle is different. For this series of tests, the excitation region was set at approximately the same size as the tank. Figure 43 shows the output of the model with an exponent of 10. The tank is enhanced, but hard to distinguish from the background, so the model was re-run with an exponent of five. This showed improvement, as can be seen in Figure 44. The program was therefore re-run with an exponent of one. The result, shown in Figure 45, shows that the model has accentuated the tank and attenuated the background.

The retina model was able to enhance objects of a certain size and shape, while attenuating other objects, in digitized images of real targets. This indicates that the model has potential to be used as a preprocessor for a target recognition system.

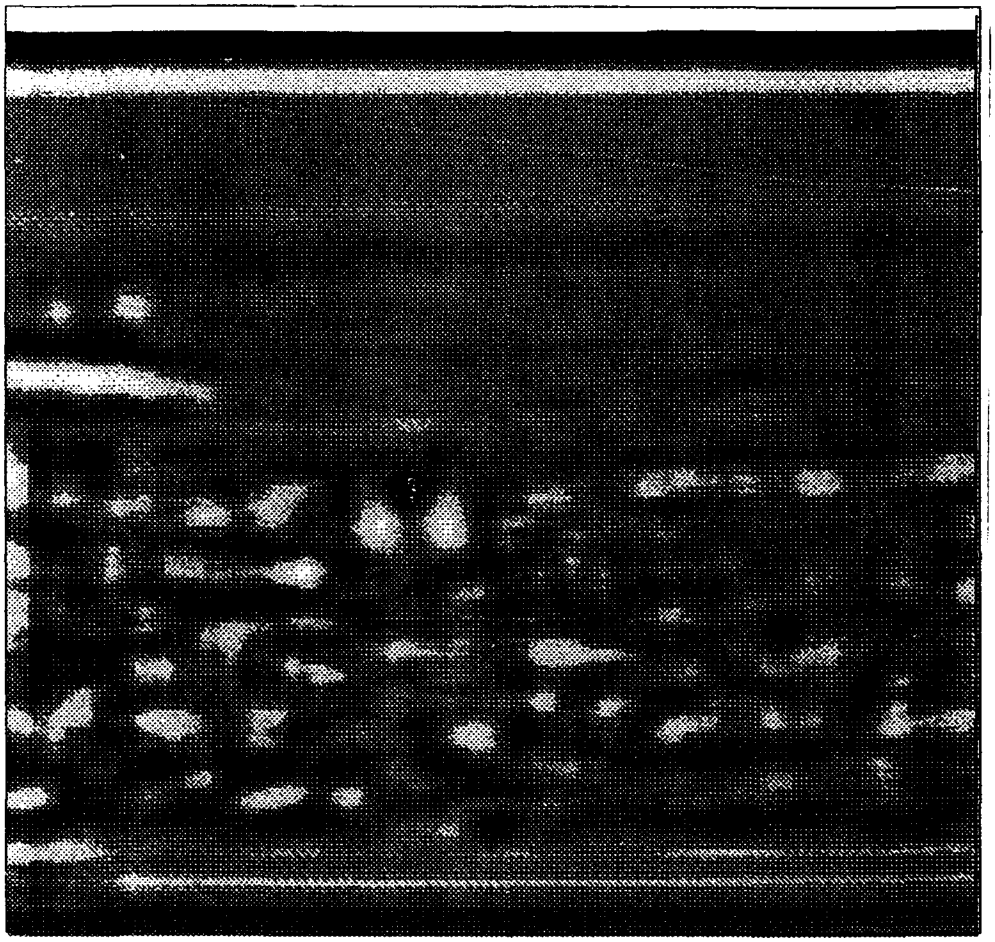


Figure 44 Case 41. Enhanced Tank,
Small Exponent

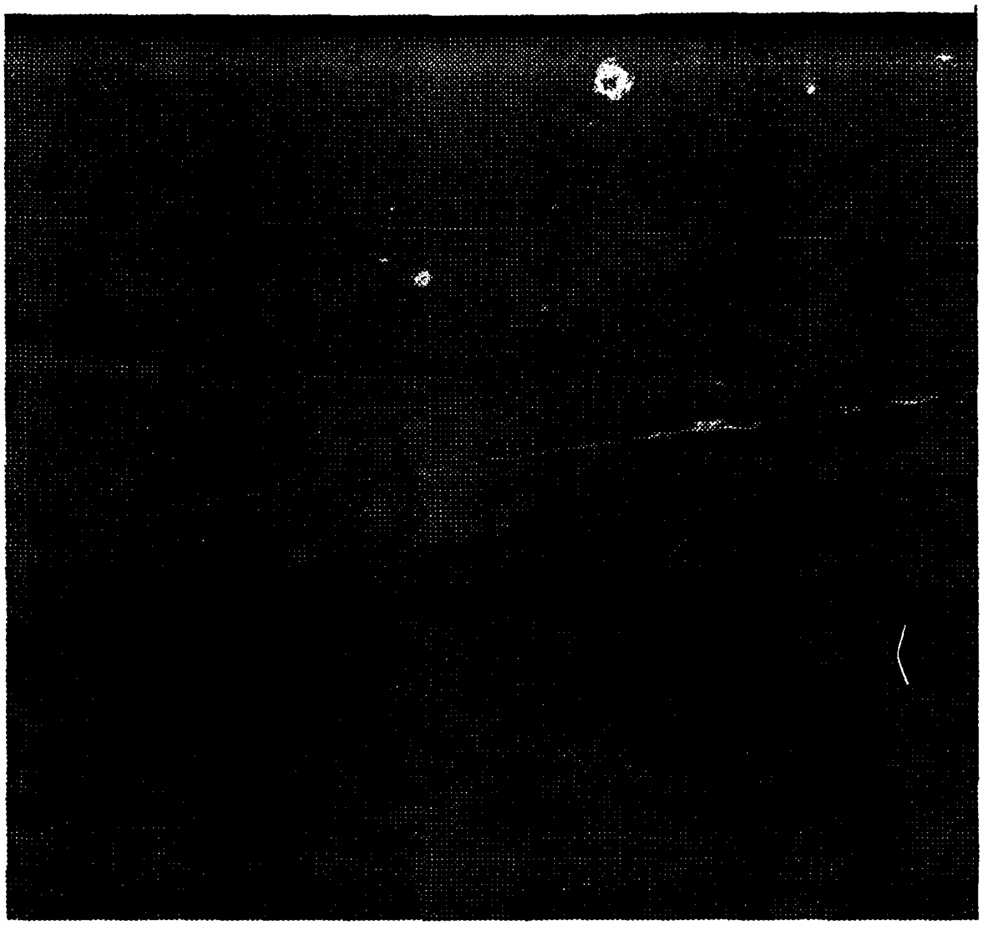


Figure 45 Case 42. Enhanced Tank, Smaller Exponent

V. Conclusion

The non-linear retinal filter implemented for this experiment can be used for edge enhancement, noise reduction, and size and shape filtering of images. In addition, since this model simulates some retinal functions, it may help understand how retinal processing is used in vertebrate visual systems. There are important applications for this type of filter.

The input control parameters are the size of the excitation region, the size of the inhibition region, and the exponent describing the steepness of the inhibition curve. The model creates a light and dark band around any intensity transition, the size of which can be changed by changing these parameters. Increasing the size of the excitation and inhibition regions widens these bands, and can also help attenuate any noise in the image. Increasing the exponent increases the magnitude of these regions, so that the difference between the high and low values of of the bands is increased. The magnitude of these transitions is also affected by the intensity values of the two areas which meet to form a transition.

When the size of the excitation region approximately matches the size of the target of interest, that target will be enhanced while other objects in the image will be attenuated. Objects much smaller than the excitation region will be minimized, while objects much larger than the excitation region will appear as edges, but the intensity of the edges will be less than the target of interest, if their original intensity values were the same.

The model provides insight into how the vertebrate retina might segment objects out of complex scenes. The vertebrate brain processes the sharpened and segmented images to make what use of the data that it can, given its phylogenetically-determined capability. The lower vertebrates' brains are small (compared to a human's), and don't have the computational power to reconstruct the visual world from optic nerve impulses. The lower vertebrate uses the filter to reduce the computational burden on its brain. Different species of vertebrate need to detect different objects to survive. This project showed how the same physiology can be used to detect different targets.

The retinal model has applications for automatic target recognition systems and for visual system research. For automatic targeting, the filter could be used to reduce the computational requirement for the subsequent image analyzer. The filter could be designed so that only potential targets are input to the analyzer rather than the entire scene. The filter could also be useful in situations where a human must identify targets in an image. Thus, it could assist a weapons operator in selecting a target from a cluttered video display of the target environment.

VI. Recommendations for Further Research

Further analysis and study of the retina model is required before it can be applied to solve practical problems. The model must be better understood before it can be optimized for a given situation.

The edge enhancement capability of the retina model should be compared to other, currently available edge enhancers such as the Gabor filter. The retinal model and other edge enhancers should be run against the same images, so the results could be compared. This would determine the usefulness of the retinal model edge detector.

The noise reduction capabilities should be quantified. A suitable metric should be developed to express the retina model's ability to smooth the noise in a given scene. Such a metric would take into account the segmentation of the target from the background, and the intensity difference between the target and background, as well as the lessening of the noise.

The size and shape filtering capability of the retina model is an area where additional research could be done. A metric should be developed to measure the "closeness" of targets, perhaps some type of measurement in 2-D Fourier space. Images with targets of varying degrees of closeness should be run, to determine which objects can be distinguished. Different levels of noise should then be added, to determine how noise can corrupt the filtering

When this size and shape filtering is fully characterized, the model should be run against more real images, using different types of targets at different aspect angles, against varying backgrounds.

In addition, it should be noted that biological retinas have several layers of lateral connections, but only one layer was modeled for this project. It is unlikely that the retina uses both layers of lateral connections for edge enhancement, since this would result in double lines around the transitions. The other layers of lateral connections should be modeled, to gain a better understanding of the purpose of these connections. Possibly, the output of this model could be binarized and, with the proper choice of threshold, be used as a mask to be superimposed on the original image. Finally, the retina model should be implemented in hardware. Once the proper parameters for a given situation have been determined, a retina filter could be fabricated. This device could preprocess images peripherally and very quickly before the images reach the a computer, just as the retina preprocesses images before the signals reach the brain.

APPENDIX

PROGRAM LISTING

```
#include <itex100.h>
#include <stdtyp.h>
#include <math>

   struct array{
          int data[512];
   };
   static struct array pic[512];
   static int  x,y,i,j,k,l,p,q;
   char filnam[64],inbuffer[64];
   int input,bksize,yextent,xextent;
   int st,xfin,yfin,jfin,ifin,xrang,yrang;
   int sigtobrain,image,nframes,num,cam;
   int change,width,height,avreg,avrang;
   int xinhibit,yinhibit;
   int loop;
   double multfactor;
   resp[512][480];
   static int pbipolarresp[512][480],bipolarresp[512][480];
/*******************************************************************************/

main()
{
  unsigned base = 0x1600;
  long mem = 0x200000L;
  int flag = 1, block = 8;
  sethdw(base,mem,flag,block);
  initialize();

  change = 0;

   for (; ;){

     printf("Deleting reserved file names\n");

     /* Deletes images acquired during */
     /* the last time through this program */

     system ("delete picture.img;*");

     printf("Enter a 1 if you want to use camera images\n");
     printf("else enter a 0 to use ready-made images\n");
     scanf("%d",&image);
     getchar();

     if (image == 1 ) {
        printf("Enter camera to use, 0,1 or 2\n");
        scanf("%d",&cam);
        getchar();
        setcamera(cam);
        printf("Enter a 1 if you want the camera\n");
        printf("to grab a selected number\n");
        printf("of successive images\n");
        printf("Enter a 0 if you want to reposition\n");
        printf("the camera after each frame for \n");
        printf("a selected number of successive images\n");
        scanf("%d",&input);
        getchar();
```

```
    printf("Enter the number of frames you want\n");
    scanf("%d",&nframes);
    getchar();
}
else nframes = 1;

printf("To avoid cutoff blocks enter a 1,2,4,8,16,or 32\n");
printf("for the blocksize (bksize) . . . . . . . . . . . . . . . . \n");
scanf("%d",&bksize);
getchar();
if (bksize < 1) bksize = 1;
printf("Enter an integer between 0 and 255 for the maximum number\n");
printf("of receptors you want to the left and right of a receptor\n");
printf("block in its excitation region\n");
scanf("%d",&xextent);
getchar();
printf("Enter an integer between 0 and 239 for the number of\n");
printf("receptor blocks you want above and below a receptor\n");
printf("block in its excitation region\n");
scanf("%d",&yextent);
getchar();
printf("How many times wider than the excitation region\n");
printf("do you want the inhibition region \n");
scanf("%d",&width);
getchar();
printf("How many times higher than the excitation region\n");
printf("do you want the inhibition region \n");
scanf("%d",&height);
getchar();
printf("Enter the region over which the average intensity\n");
printf("will be calculated for a cell and assigned to a cell\n");
scanf("%d",&avreg);
getchar();
printf("Enter the exponent corresponding to the steepness\n");
printf("of the retinal curve\n");
scanf("%f",&multfactor);
getchar();
printf("Enter the 1/2 size of the inhibition region - horizontal\n");
scanf("%d",&xinhibit);
getchar();
printf("Enter the 1/2 size of the inhibition region - vertical\n");
scanf("%d",&yinhibit);
getchar();

xrang = xextent * bksize;
yrang = yextent * bksize;
avrang = avreg * bksize;
st = (int)(bksize/2);
xfin = 512;
yfin = 480;

if (image == 1)  savecameraimages();    /* Acquires camera images */
                                        /* and saves them on disk */

else displaystoredimage();              /* The user can select one */
                                        /* of 3 prestored images and */
                                        /* display it on the monitor */

for (y = 0; y < 480; y++) {
  for (x = 0; x < 512; x++) {
         pbipolarresp[x][y] = 0;                /* Previous bipolar response */
     ) )                                        /* Sleeping state */


     for (num = 1; num < nframes+1; num++) (
       if (image == 1) (
                                                /* Retrieves from disk memory a */
                                                /* camera image and displays it */
                                                /* on the monitor screen */

         readim(0,0,512,480,"picture.img;num","no");
         for (y = 0; y <480; y++) (
           rhline(0,y,512,pic[y].data);         /* Image is saved in an array */
         )
       )


       if (bksize != 1) (      /* Uses original scene */
       create();               /* This function creates bksize x bksize blocks */
       )                       /* of pixels(Each block represents 1 cell) */
                               /* and assigns to each pixel in the block the */
                               /* average intensity value of the block */
       if (bksize ==1) (
         average();            /* Calculates the average for a region around */
                               /* a cell block that will be inhibited and */
                               /* assigns this intensity to the cell block */
       )

         inhibit();            /* Output from a cell block */
     )

/*******************************************************************************/
/*** Save Image in Video Frame Memory to Disk***/

      filnam[0] = '\0';
      strcat(filnam,"DUA2:[srogers.jbrickey.images]\0");
      printf("\n Entire screen will be saved as an 8-bit image \n");
      printf("Enter file name with 3 char suffix, i.e. .img \n\n >");
      scanf("%s",inbuffer);
      strcat(filnam,inbuffer);
      printf("\n\n\n\n Saving image...");
      saveim(0,0,511,480,0,filnam,"NOCOMM");

/*******************************************************************************/

     printf("Enter a 1 if you want to rerun the program\n");
     printf("otherwise enter a 0\n");
     scanf("%c",&input);
     getchar();
     if (input == 0) (
       break;
     )
   )
)
/*********************************************************************************/
displaystoredimage()
(

for(; ;)(
    printf("Enter a 0 if you want to display a rectangle\n");
    printf("Enter a 1 if you want to display a noisy scene\n");
    printf("Enter a 2 if you want to display vertical bars\n");
    printf("Enter a 3 if you want pic.pic (half lopass/half orig) \n");
    printf("with three low intensity rectangles\n");
    scanf("%d",&input);
    getchar();

    if (input != 0 && input != 1 && input != 2 && input != 3){
      printf("Bad input\n");
    }
    else {
      break;
      printf("break\n");
    }
  }
    if (input == 0){
      sclear(55,1);                              /*Clears the screen*/
      aclear(248,32,144,144,25);                 /*Displays a rectangle*/
      printf("Okay .  Reading image into array\n");
      for (y = 0; y <480; y++) {
        rhline(0,y;512,pic[y].data);
      }
    }

    if (input == 1){
      readim(0,0,512,480,"block.pic","nc");    /*Reads in the noisy scene*/
      printf("Okay .  Reading image into array\n");
      for (y = 0; y <480; y++) {
        rhline(0,y,511,pic[y].data);
      }
    }

    if (input == 2){                  /* Displays vertical bars of increasing */
                                      /* intensity across the monitor. */
      aclear(0,0,64,480,64);
      aclear(64,0,64,480,80);
      aclear(128,0,64,480,96);
      aclear(192,0,64,480,112);
      aclear(256,0,64,480,128);
      aclear(320,0,64,480,144);
      aclear(384,0,64,480,160);
      aclear(448,0,64,480,176);
      printf("Okay .  Reading image into array\n");
      for (y = 0; y <480; y++) {
        rhline(0,y,512,pic[y].data);
      }
    }
  return;

  }
  /**************************************************************************/
  savecameraimages()               /* Displays successive images on the screen */
  {

  if (input == 1) {
    for (num = 1; num < nframes-1; num++) {
      snap(1);                                    /* Takes (n) consecutive */
      saveim(0,0,512,480,0,"picture.img","nc");   /* pictures and saves them *
```

```
)
else if (input == 0) {
  for (num = 1; num < nframes+1; num++) {
    snap(1);                                        /* Takes a picture */
    saveim(0,0,512,480,0,"picture.img","no");       /* and saves it */
    if (num < nframes) {
       printf("Reposition the camera if you so desire\n");
       printf("After the camera has been properly\n");
       printf("positioned hit the CR key\n");
       getchar();
    }
  }
}

return;

}
/*****************************************************************************/
average()

{
int xlc,xrc,yuc,ydc;
int sumint,nr;

printf("subroutine  AVERAGE\n");
printf("processing. please wait...\n");

for (y = st; y < yfin; y+=bksize) {
  for (x = st; x < xfin; x+=bksize) {

    if (x < avrang+st) xlc = st;
    else xlc = x-avrang;
                                         /* Calculates the left-most x */
                                         /* coordinate of the excitation */
                                         /* region */

    if (x > 512-avrang-st) xrc = 512-st;
    else xrc = x+avrang;
                                         /*Calculates the right-most x */
                                         /* coordinate of the excitation */
                                         /* region */

    if (y < avrang+st) ydc = st;
    else ydc = y-avrang;
                                         /* Calculates the lowest value */
                                         /* of y for the excitation region */
                                         /* (Above yuc on the screen) */

    if (y > 480-avrang-st) yuc = 480-st;
    else  yuc = y+avrang;
                                         /* Calculates the highest value */
                                         /* of y for the excitation region */
                                         /* (Below ydc on the screen) */
    nr = 0;
    sumint = 0;

    jfin = yuc-bksize;
    ifin = xrc-bksize;

    for (j = ydc; j < jfin; j+=bksize) {    /* Calculates the average */
```

```
          for (i = xlc; i < ifin; i+=bksize) {       /* value for a region */
                                                     /* input by the user  */
            nr = nr + 1;
            sumint = pic[j].data[i]+sumint;

        } }

        resp[x][y] = (int)(sumint/nr);
    } }

    /* Displays the original scene after being averaged */

    for (y =0; y < 480; y++) {
      for (x=0; x < 512; x++) {
        pic[y+st].data[x+st] = resp[x+st][y+st];
        aclear(x,y,bksize,bksize,pic[y+st].data[x+st]);
    } }
    printf("ok finished averaging\n");

    return;

    }
    /******************************************************************************/
    create()                       /* This function creates bksize x bksize blocks */
                                   /* of pixels(Each block represents 1 retinal) */
                                   /* cell and assigns each pixel in a block the */
                                   /* average intensity value of the block */

    {
    int  count,sum,avgint,xstart,ystart;
    count=sum=0;

    printf("Creating blocks representing receptor cells . . .\n");

    for(l = 0; l < 480; l+=bksize) {

                                     /* This loop reads 30 vertical blocks */
                                     /* of bksize x bksize pixels(receptor cell) */

      for (k = 0; k < 512; k+=bksize){

                                     /* This loop reads 32 horizontal blocks */
                                     /* of bksize x bksize pixels(receptor cell) */

      /* Calculates the blocksize(1 block represents 1 receptor cell) */

        for (j = 0; j < bksize; j++) {
                                            /* This loop reads bksize lines */
                                            /* of bksize pixels */

          for (i = 0; i < bksize; i++){

                                      /* This loop reads 1 line of bksize pixels */

            count = count + 1;

            if (count == 1) {         /* If count = 1, save the upper left-hand */
              xstart = i+k;           /* x and y coordinates of the block */
              ystart = j+l;
          sum += pic[j+1].data[i+k];       /* Sums up the intensities in the */
                                           /* bksize x bksize block */
      } }

      count = 0;

      avgint = (int) (sum/(bksize*bksize));

                                           /* Calculates the average intensity */
                                           /* for the bksize x bksize block */
                                           /* of pixels */

      sum = 0;

      for (j = ystart;j < ystart+bksize; j++) {

                                           /* Fills the bksize x bksize block */
                                           /* of pixels with the average*/
                                           /* intensity value of the block */

        for (i = xstart; i < xstart+bksize; i++){

          pic[j].data[i] = avgint;         /* Stores the average value for each */
                                           /* bksize x bksize block */
                                           /* (receptor cell) */

      } }
  } }

   for(y = 0; y < 480; y++){                /* Displays on the monitor */
     whline(0,y,512,pic[y].data);           /* receptor blocks */
   }                                        /* Each block represents */
                                            /* 1 receptor cell */
  return;

  }
  /**************************************************************************************/
  inhibit()                      /* This subroutine calculates the response */
                                 /* of each bipolar cell block and ganglion */
                                 /* cell block (sustained and transient types */
                                 /* to illumination in the scene. */

  {
  int xlc,xrc,yuc,ydc;
  int exlc,exrc,exyuc,exydc;
  int loop1,loop2,loop3,loop4;
  int suminhibreg,sumextreg;
  int nrextreg,nrinhibreg;
  double k9,averageint,koveri,ratio;

  printf("processing. please wait...\n");

  for (y = st; y < yfin; y+=bksize) {
    for (x = st; x < xfin; x+=bksize) {

      if (x < xrang-st) xlc = st;
      else xlc = x-xrang;
                                          /* Calculates the left-most x */
                                          /* coordinate of the excitation */
                                          /* region */
```

```
if (x > 512-xrang-st) xrc = 512-st;
else xrc = x+xrang;
                                  /*Calculates the right-most x */
                                  /* coordinate of the excitation */
                                  /* region */

if (y < yrang+st) ydc = st;
else ydc = y-yrang;
                                  /* Calculates the lowest value */
                                  /* of y for the excitation region */
                                  /* (Above yuc on the screen) */

if (y > 480-yrang-st) yuc = 480-st;
else  yuc = y+yrang;
                                  /* Calculates the highest value */
                                  /* of y for the excitation region */
                                  /* (Below ydc on the screen) */

if (xlc != st) {                  /* The inhibition region exists */
  if (xlc-xinhibit < st) {   /* to the left of xlc */
    exlc = st;
    loop1 = exlc;
  }
   else {
     exlc = xlc-xinhibit;
     loop1 = exlc;
   }
}
else loop1 = xlc;                   /* No inhibition region exists */
                                    /* to the left of xlc */

if (xrc != 512-st) {                /* The inhibition region exists */
  if (xrc+xinhibit > 512-st) {      /* to the right of xrc */
    exrc = 512-st;
    loop2 = exrc;
  }

  else {
    exrc = xrc+xinhibit;
    loop2 = exrc;
  }
}
else {
  loop2 = xrc;                       /* The inhibition region does */
}                                    /* not exist to the right of xrc */


if (ydc != st) {                     /* The inhibition region exists */
  if (ydc-yinhibit < st) {           /* below ydc */
    exydc = st;
    loop3 = exydc;
  }

  else {
    exydc = ydc-yinhibit;
    loop3 = exydc;
  }
}
else {
    loop3 = ydc;                          /* The inhibiton region does */
  }                                       /* not exist below ydc */


  if (yuc != 480-st) {                    /* The inhibition region exists */
    if (yuc+yinhibit > 480-st) {          /* above yuc */
      exyuc = 480-st;
      loop4 = exyuc;
    }

    else {
      exyuc = yuc+yinhibit;
      loop4 = exyuc;
    }
  }
  else {
    loop4 = yuc;                           /* The inhibition region does not */
  }                                        /* exist above yuc */


  nrextreg = 0;
  nrinhibreg = 0;
  suminhibreg = 0;
  sumextreg = 0;

  ifin = loop2+bksize;
  jfin = loop4+bksize;


  for (j = loop3; j < jfin; j+=bksize) {
    for (i = loop1; i < ifin; i+=bksize) {
      if((i >= xlc) && (i <= xrc)&&
          (j >= ydc) && (j <= yuc)) {

                                              /* Checks to see if */
                                              /* a cell is excitatory */

        nrextreg = nrextreg + 1;              /* Calculates the number of */
                                              /* of excitatory cells. */

        sumextreg = pic[j].data[i]+sumextreg;

                                              /* Sums up the intensities */
      }                                       /* in the excitory region */
      else {

        nrinhibreg = nrinhibreg - 1;          /* Calculates the number of */
                                              /* cells in the excitory region */

        suminhibreg = pic[j].data[i]-suminhibreg;

                                              /* Sums up the intensities */
                                              /* in the inhibitory region */
      }

  } }

  .verageint = ((double)(sumextreg))/((double)(nrextreg));

                                           /* Calculates the average intensity */
    k9 = ((double)(suminhibreg))/((double)(nrinhibreg));
                                               /* Calculates the average intensity */
                                               /* value for the inhibitory region */


/* Calculates the output of a cell block */

    averageint += 1.0;
    ratio = pow(averageint,multfactor)/(pow(averageint,multfactor) +
                pow(k9,multfactor));

    if (change == 1) {
    /* resp[x][y] = (int) (128.0*ratio); */
    }
    else {
      resp[x][y] = (int) (255.0*ratio);
      if(resp[x][y] > 255) resp[x][y] = 255;
      if(resp[x][y] < 0) resp[x][y] = 0;
    }

}}

/* Displays the transient response of each cell block */

printf("Press CR to display inhibited image/n");
getchar();
for (p = 0; p < 480; p+=bksize) {
  for (q = 0; q < 512; q+=bksize) {
    aclear(q,p,bksize,bksize,resp[q+st][p+st]);
} }

printf("\n ok finished inhibit");

return;

}
```

BIBLIOGRAPHY

1. Andrews, Harry C. and Charles F. Hall. Digital Color Image Compression in a Perceptual Space. AFAL-TR-78-80. University of Southern California, Los Angles CA. Contract Number F33615-77-C-1016 with Air Force Wright Aeronautical Laboratories (AFSC), Wright-Patterson AFB OH. June 1978

2. Kabrisky, Matthew, PhD. Lectures given in EENG 617, 618, 619, Sensory Information Processing Sequence. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH. September 1986 through June 1987.

3. Sillart, Capt Jeffrey E. A Computer Model of Inhibition. Energy Normalization, and Noise Suppression in the Retina. MS Thesis, AFIT/GE/ENG/87D-59, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH. December 1987.

4. Werblin, Frank S. "The Control of Retinal Sensitivity," Journal of General Physiology, 63: 62-67 (1974)

5. -------, Frank S. "The Control of Sensitivity in the Retina," Scientific American, 228: 70-79 (January 1973)

VITA

Stephen M. Wiener [redacted] He graduated from Matawan Regional High School, New Jersey, in 1978. He then attended Case Western Reserve University, from which he received a Bachelor of Science in Electrical Engineering degree in 1982. Following graduation, he went to Officer Training School at Lackland AFB, Texas, and received a commission in the USAF. He served as an Armament Systems Analyst at the Foreign Technology Division at Wright-Patterson AFB, Ohio, before entering the Air Force Institute of Technology in 1986. He is currently stationed at Eglin AFB, Florida.

[redacted]

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION ' OWNGRADING SCHEDULE

3 DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4 PERFORMING ORGA..IZATION REPORT NUMBER(S)
AFIT/GE/ENG/90S-01

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b OFFICE SYMBOL *(If applicable)*
AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433-6583

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING/SPONSORING ORGANIZATION

8b OFFICE SYMBOL *(If applicable)*

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO

11. TITLE *(Include Security Classification)*
See Box 19

12. PERSONAL AUTHOR(S)
Stephen M. Wiener, BSEE, Captain, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT *(Year, Month, Day)*
1990 August

15. PAGE COUNT
92

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
06	04	
17	08	

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Retina, Edge Enhancement, Noise Reduction, Image Processing, Signal Processing

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Title: ANALYSIS OF A VERTEBRATE RETINA MODEL

Thesis Chairman: Matthew Kabrinsky
Professor of Electrical Engineering

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL
Matthew Kabrinsky

22b TELEPHONE *(Include Area Code)*
513-255-9267

22c OFFICE SYMBOL
AFIT/ENG

Previous editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Block #19

ABSTRACT

This thesis analyzed a computer model of the excitation-inhibition system in a generic vertebrate retina. This model enhanced edges, eliminated brightness variations, and attenuated noise in an image. Input parameters were changed to determine their effect on the model's properties. The model was then used as a size and shape filter. Depending on the selection of input parameters, certain objects in a scene would be enhanced while others would be attenuated.